

(19) 日本国特許庁 (J P)

## (12) 公表特許公報 (A)

(11) 特許出願公表番号

特表2002-514813

(P2002-514813A)

(43) 公表日 平成14年5月21日 (2002.5.21)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テマコード* (参考)
G 0 6 F 13/00	3 0 1	G 0 6 F 13/00	3 0 1 Q 5 B 0 1 4
11/14	3 1 0	11/14	3 1 0 F 5 B 0 2 7
11/20	3 1 0	11/20	3 1 0 A 5 B 0 3 4
12/00	5 1 4	12/00	5 1 4 E 5 B 0 8 2
	5 4 5		5 4 5 A 5 B 0 8 3

審査請求 未請求 予備審査請求 有 (全 70 頁) 最終頁に続く

(21) 出願番号 特願2000-548806 (P2000-548806)  
 (86) (22) 出願日 平成11年5月7日 (1999.5.7)  
 (85) 翻訳文提出日 平成12年11月10日 (2000.11.10)  
 (86) 国際出願番号 PCT/US99/09903  
 (87) 国際公開番号 WO99/59064  
 (87) 国際公開日 平成11年11月18日 (1999.11.18)  
 (31) 優先権主張番号 09/076, 388  
 (32) 優先日 平成10年5月12日 (1998.5.12)  
 (33) 優先権主張国 米国 (US)  
 (31) 優先権主張番号 09/076, 347  
 (32) 優先日 平成10年5月12日 (1998.5.12)  
 (33) 優先権主張国 米国 (US)

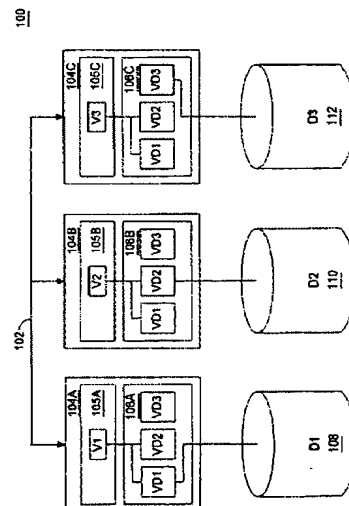
(71) 出願人 サン・マイクロシステムズ・インコーポレ  
 ーテッド  
 SUN MICROSYSTEMS, IN  
 CORPORATED  
 アメリカ合衆国 94303 カリフォルニア  
 州・パロ アルト・サン アントニオ ロ  
 ード・901  
 (72) 発明者 スローター, グレゴリー・エル  
 アメリカ合衆国・94306・カリフォルニア  
 州・パロ アルト・エマーソン ストリー  
 ト・3326  
 (74) 代理人 弁理士 山川 政樹

最終頁に続く

(54) 【発明の名称】 高可用性クラスタ仮想ディスク・システム

## (57) 【要約】

クラスタは、そのクラスタの各記憶装置へのアクセス権をそのクラスタの各ノードに与える仮想ディスク・システムを実現する。この仮想ディスク・システムは、障害が存在する状態で記憶装置にアクセスすることができ、データ・アクセス要求が確実に完了するような高い可用性を有する。ノード間で一貫したマッピングおよびファイル許可データを保証するために、データは高可用性クラスタ・データベースで記憶される。クラスタ・データベースは障害が存在する状態でもノードに一貫したデータを提供するので、各ノードは一貫したマッピングおよびファイル許可データを有することになる。ノード間のリンクを確立し、そのリンクを管理するクラスタ・トランスポート・インタフェースが提供される。クラスタ・トランスポート・インタフェースが受け取るメッセージは1つまたは複数のリンクを介して宛先ノードに運搬される。クラスタの構成は動作中に変更することができる。構成を変更する前に、再構成手順はデータ・アクセス要求を中断し、保留中のデータ・アクセス要求が完了するのを待つ。新しい構成を反映するために、再構成が



**【特許請求の範囲】**

**【請求項 1】** 第 1 のノードと、第 2 のノードと、前記第 1 のノードと前記第 2 のノードとの間に結合された通信リンクと、

前記第 1 のノードに結合された記憶装置であって、前記記憶装置が前記第 2 のノードに結合されているように見えるように構成された仮想ディスク・システムを使用して前記記憶装置にアクセスするように前記第 2 のノードが構成される記憶装置とを含み、

前記仮想ディスク・システムが前記第 2 のノード上で動作するドライバと前記第 1 のノード上で動作するマスタとを含み、前記第 2 のノードが前記仮想ディスク・システムの仮想ディスクにアクセスすると、前記ドライバは前記マスタにデータ要求を送るように構成され、前記マスタは前記記憶装置からのデータにアクセスするように構成され、前記マスタは前記通信リンクを介して前記ドライバに応答を送るように構成され、

前記ドライバは前記ドライバが前記応答を受け取るまで前記データ要求のコピーを記憶するように構成され、前記ドライバは前記ドライバが前記応答を受け取り損なった場合に前記要求を再送するように構成される、分散コンピューティング・システム。

**【請求項 2】** 前記データ通信インタフェースおよび前記記憶装置に結合された第 3 のノードをさらに含み、前記ドライバは前記ドライバが前記応答を受け取り損なった場合に前記第 3 のノード上の第 2 のマスタに前記データ要求を再送するように構成され、前記第 2 のマスタは前記記憶装置に関するデータにアクセスするように構成される、請求項 1 に記載の分散コンピューティング・システム。

**【請求項 3】** 前記ドライバが前記第 1 のノードまたは通信リンクの障害のために前記データを受け取り損なった場合、前記ドライバは前記分散コンピューティング・システムが再構成したあとで前記データ要求を再送するように構成される、請求項 1 に記載の分散コンピューティング・システム。

**【請求項 4】** 前記再送データ要求が、前記記憶装置に結合された第 3 のノードに送られる、請求項 3 に記載の分散コンピューティング・システム。

【請求項5】 前記仮想ディスク・システムが前記記憶装置と通信するために一次ノードと代替ノードとを維持し、前記第2のノードは前記第2のノードが応答を受け取り損なった場合に前記一次ノードに前記データ要求を送り、前記代替ノードに前記データ要求を再送するように構成される、請求項4に記載の分散コンピューティング・システム。

【請求項6】 前記第1のノードが前記一次ノードであり、前記第3のノードが前記代替ノードである、請求項5に記載の分散コンピューティング・システム。

【請求項7】 第1のノードと、第2のノードと、前記第1のノードと前記第2のノードとの間に結合された通信リンクと、

前記第1のノードに結合された記憶装置であって、前記記憶装置が前記第2のノードに結合されているように見えるように構成された仮想ディスク・システムを使用して前記記憶装置にアクセスするように前記第2のノードが構成される記憶装置とを含み、

前記仮想ディスク・システムは前記記憶装置に対する仮想ディスクのマッピングを含むように構成され、前記第1のノードと前記第2のノードは一貫したマッピング・データを受け取るように構成される、分散コンピューティング・システム。

【請求項8】 前記マッピングが、前記記憶装置に結合されたノードと、前記記憶装置に対応する前記ノード上のディスク装置とを識別し、前記マッピングが前記第1のノードおよび前記第2のノードによってアクセスされる高可用性データベースに記憶される、請求項7に記載の分散コンピューティング・システム。

【請求項9】 前記高可用性データベースがクラスタ構成データベースである、請求項8に記載の分散コンピューティング・システム。

【請求項10】 前記マッピングが、前記記憶装置に結合された一次ノードと前記記憶装置に結合された代替ノードとを示すデータを含む、請求項7に記載の分散コンピューティング・システム。

【請求項11】 データ・アクセス要求は前記マッピングが更新されたとき

に中断されるように構成される、請求項7に記載の分散コンピューティング・システム。

【請求項12】 前記マッピングは、ノードがクラスタに加わるかまたはクラスタを離れるときに更新されるように構成される、請求項11に記載の分散コンピューティング・システム。

【請求項13】 データ・アクセス要求が再開されたときにノードが新しいマッピングを求めてデータベースに照会する、請求項12に記載の分散コンピューティング・システム。

【請求項14】 動作中に分散コンピューティング・システムの構成を更新することができる、請求項13に記載の分散コンピューティング・システム。

【請求項15】 第1のノードと、第2のノードと、前記第1のノードと前記第2のノードとの間に結合された通信リンクと、

前記第1のノードに結合された記憶装置であって、前記記憶装置が前記第2のノードに結合されているように見えるように構成された仮想ディスク・システムを使用して前記記憶装置にアクセスするように前記第2のノードが構成される記憶装置とを含み、

前記記憶装置の許可データが前記第1のノードと前記第2のノードとの間で一貫している、分散コンピューティング・システム。

【請求項16】 前記許可データは、前記第1のノードおよび前記第2のノードによってアクセスされる高可用性データベースに記憶されるように構成される、請求項15に記載の分散コンピューティング・システム。

【請求項17】 前記高可用性データベースがクラスタ構成データベースである、請求項16に記載の分散コンピューティング・システム。

【請求項18】 第1のノードと、第2のノードと、前記第1のノードと前記第2のノードとの間に結合された通信リンクと、

前記第1のノードに結合された第1の記憶装置であって、前記第1の記憶装置が前記第2のノードに結合されているように見えるように構成された仮想ディスク・システムを使用して前記第1の記憶装置にアクセスするように前記第2のノードが構成される第1の記憶装置と、

前記第2のノードに結合された第2の記憶装置であって、前記第2の記憶装置が前記第1のノードに結合されているように見えるように構成された仮想ディスク・システムを使用して前記第2の記憶装置にアクセスするように前記第1のノードが構成される第2の記憶装置と、

前記仮想ディスク・システムの上または前記仮想ディスク・システムの下に層状に重ねたボリューム・マネージャを含む、分散コンピューティング・システム。

【請求項19】 前記ボリューム・マネージャが前記仮想ディスク・システムの下に層状に重ねられ、前記仮想ディスク・システムはボリュームにアクセスするように構成され、前記ボリュームは前記第1の記憶装置または前記第2の記憶装置にアクセスするように構成される、請求項18に記載の分散コンピューティング・システム。

【請求項20】 前記ボリュームが前記仮想ディスク・システムにとって記憶システムとして現れる、請求項19に記載の分散コンピューティング・システム。

【請求項21】 前記ボリュームが高可用性記憶装置である、請求項20に記載の分散コンピューティング・システム。

【請求項22】 前記ボリューム・マネージャが前記仮想ディスク・システムの上に層状に重ねられ、クライアントが前記仮想ディスク・システムの仮想ディスクにアクセスするように構成されたボリュームにアクセスすることができる、請求項18に記載の分散コンピューティング・システム。

【請求項23】 前記ボリュームは2つまたはそれ以上の仮想ディスクにアクセスするように構成される、請求項22に記載の分散コンピューティング・システム。

【請求項24】 第1のノードと、第2のノードと、前記第1のノードと前記第2のノードとの間に結合された通信リンクと、

前記第1のノードに結合された記憶装置であって、前記記憶装置が前記第2のノードに結合されているように見えるように構成された仮想ディスク・システムを使用して前記記憶装置にアクセスするように前記第2のノードが構成される記

憶装置とを含み、

前記仮想ディスク・システムが前記第2のノード上で動作するドライバと前記第1のノード上で動作するマスタとを含み、前記第2のノードが前記記憶装置に対応する仮想ディスクにアクセスすると、前記ドライバは前記マスタにデータ要求を送るように構成され、前記マスタは前記記憶装置からのデータにアクセスするように構成され、前記マスタは前記通信インタフェースを介して前記ドライバに前記データを送るように構成され、

前記ドライバは前記ドライバが前記データを受け取るまで前記データ要求のコピーを記憶するように構成され、前記ドライバは前記ドライバが前記データを受け取り損なった場合に前記データ要求を再送するように構成され、前記仮想ディスク・システムは前記記憶装置に対する仮想ディスクのマッピングを含むように構成され、前記第1のノードと前記第2のノードはノード障害が発生した場合に一貫したマッピング・データを受け取るように構成され、前記記憶装置の許可データが前記第1のノードと前記第2のノードとの間で一貫している、分散コンピューティング・システム。

【請求項25】 前記データ通信インタフェースおよび前記記憶装置に結合された第3のノードをさらに含み、前記ドライバは前記ドライバが前記データを受け取り損なった場合に前記第3のノードに前記データ要求を再送するように構成される、請求項24に記載の分散コンピューティング・システム。

【請求項26】 前記マッピングが、前記記憶装置に結合された一次ノードと前記記憶装置に結合された二次ノードとを示すデータを含む、請求項25に記載の分散コンピューティング・システム。

【請求項27】 前記マッピングおよび前記許可データは、前記第1のノードおよび前記第2のノードによってアクセス可能な高可用性データベースに記憶されるように構成される、請求項26に記載の分散コンピューティング・システム。

【請求項28】 第1のノードと、第2のノードと、前記第1のノードおよび前記第2のノードに結合された通信リンクとを含み、前記第1のノードおよび前記第2のノードが前記記憶装置にアクセスし、

前記記憶装置が前記記憶装置に関連する許可データを有し、前記許可データが前記第1のノードおよび前記第2のノードによってアクセス可能な高可用性分散データベースに記憶され、特定のノードが前記記憶装置をオープンすると、前記特定のノードが前記記憶装置に関する前記許可データによって装置ファイルを作成し、それにより、ノード障害が存在する状態で前記第1のノードと前記第2のノードが一貫した許可データを入手する、分散コンピューティング・システム。

【請求項29】 前記許可データが、所有者と、グループと、前記所有者および前記グループのための許可モードとを含む、請求項28に記載の分散コンピューティング・システム。

【請求項30】 前記許可モードが、読取りと、書込みと、実行とを含む、請求項29に記載の分散コンピューティング・システム。

【請求項31】 前記高可用性データベースがクラスタ構成データベースである、請求項28に記載の分散コンピューティング・システム。

【請求項32】 前記記憶装置がディスク装置である、請求項28に記載の分散コンピューティング・システム。

【請求項33】 前記第1のノードが前記記憶装置に直接アクセスし、前記第2のノードが前記通信リンクを介して前記記憶装置にアクセスする、請求項28に記載の分散コンピューティング・システム。

【請求項34】 前記記憶装置が特定のノードによって最初にオープンされたときに前記装置ファイルが作成される、請求項28に記載の分散コンピューティング・システム。

【請求項35】 記憶装置を含む分散コンピューティング・システム内の複数のノード間で一貫した許可データを維持する方法であって、

高可用性分散データベースに前記許可データを記憶するステップと、

前記複数のノードのうちの第1のノードが装置をオープンし、前記高可用性データベースにアクセスして前記装置に関する許可データを入手するステップと、

前記複数のノードのうちの第2のノードが前記装置をオープンし、前記高可用性データベースにアクセスして前記装置に関する前記許可データを入手するステップとを含み、

それにより、障害が存在する状態で前記第1のノードと前記第2のノードが一貫した許可データを入手する方法。

【請求項36】 前記装置をオープンする前記ノードが前記記憶装置に関する装置ファイルを作成し、前記装置ファイルが前記許可データを含む、請求項35に記載の方法。

【請求項37】 前記許可データが、所有者と、グループと、前記所有者および前記グループのための許可モードとを含む、請求項36に記載の方法。

【請求項38】 前記許可モードが、読取りと、書込みと、実行とを含む、請求項37に記載の方法。

【請求項39】 前記高可用性データベースがクラスタ構成データベースである、請求項35に記載の方法。

【請求項40】 前記記憶装置がディスク装置である、請求項35に記載の方法。

【請求項41】 前記第1のノードが前記記憶装置に直接アクセスし、前記第2のノードが前記データ通信リンクを介して前記記憶装置にアクセスする、請求項35に記載の方法。

【請求項42】 前記記憶装置が特定のノードによって最初にオープンされたときに前記装置ファイルが作成される、請求項36に記載の方法。

【請求項43】 分散コンピューティング・システム内の複数のノード間で一貫した許可データを維持するためのプログラム命令を含むコンピュータ可読記憶媒体であって、前記プログラム命令が前記分散コンピューティング・システムの前記複数のノード上で実行され、前記プログラム命令が、

高可用性コヒーレント・データベースに前記許可データを記憶するステップと、

前記複数のノードのうちの第1のノードが装置をオープンし、前記高可用性データベースにアクセスして前記装置に関する許可データを入手するステップと、

前記複数のノードのうちの第2のノードが前記装置をオープンし、前記高可用性データベースにアクセスして前記装置に関する前記許可データを入手するステップとを実施するように動作可能であり、



それにより、障害が存在する状態で前記第1のノードと前記第2のノードが一貫した許可データを入手する、コンピュータ可読記憶媒体。

【請求項44】 前記ファイルが前記記憶装置に関するものであり、前記装置ファイルが前記許可データを含む、請求項43に記載の媒体。

【請求項45】 前記許可データが、所有者と、グループと、前記所有者および前記グループのための許可モードとを含む、請求項44に記載の媒体。

【請求項46】 前記許可モードが、読取りと、書込みと、実行とを含む、請求項45に記載の媒体。

【請求項47】 前記高可用性データベースがクラスタ構成データベースである、請求項43に記載の媒体。

【請求項48】 前記記憶装置が特定のノードによって最初にオープンされたときに前記装置ファイルが作成される、請求項44に記載の媒体。

【請求項49】 1つの通信リンクに結合された1つまたは複数のノードであって、1つまたは複数の装置を含む1つまたは複数のノードと、前記1つまたは複数のノードに結合された1つまたは複数の記憶装置と、

前記1つまたは複数のノードによってアクセス可能な高可用性データベースであって、障害が存在する状態で前記1つまたは複数のノードにコヒーレント・データを提供する高可用性データベースとを含み、

前記1つまたは複数の記憶装置に対する前記1つまたは複数の装置のマッピングが前記高可用性データベースに記憶され、

前記マッピングが更新されると、前記高可用性データベースが前記マッピングを更新する前に前記1つまたは複数のノードが前記1つまたは複数の記憶装置へのデータ要求の発行を停止し、前記マッピングが更新されたときに前記1つまたは複数のノードがデータ要求の発行を再開する、分散コンピューティング・システム。

【請求項50】 前記ノードがデータ要求の発行を停止したときに前記ノードが前記高可用性データベースに肯定応答信号を送る、請求項49に記載の分散コンピューティング・システム。

【請求項51】 前記1つまたは複数のノードが、前記肯定応答信号を送る

前に未処理のデータ要求が完了するのを待つ、請求項50に記載の分散コンピューティング・システム。

【請求項52】 前記高可用性データベースが、前記肯定応答信号を受け取ったあとで前記マッピングを更新する、請求項51に記載の分散コンピューティング・システム。

【請求項53】 前記高可用性データベースが第1の同期信号を出力して、前記マッピングの保留中の更新を示す、請求項52に記載の分散コンピューティング・システム。

【請求項54】 前記高可用性データベースが第2の同期信号を出力して、前記マッピングが更新されることを示す、請求項53に記載の分散コンピューティング・システム。

【請求項55】 前記第1の同期コマンドと前記第2の同期コマンドが前記1つまたは複数のノードに同時に発行される、請求項54に記載の分散コンピューティング・システム。

【請求項56】 前記高可用性データベースが、前記マッピングを更新する前に各アクティブ・ノードからの肯定応答信号を待つ、請求項55に記載の分散コンピューティング・システム。

【請求項57】 前記コンピュータ・システムが、データを失うかまたは破損することなしに前記1つまたは複数のノードと前記1つまたは複数の記憶装置との間の前記接続を再構成する、請求項56に記載の分散コンピューティング・システム。

【請求項58】 記憶装置に対するノードのマッピングを更新する方法であって、

前記マッピングを高可用性データベースに記憶するステップであって、前記データベースが前記ノードによってアクセス可能であり、障害が存在する状態で前記ノードにコヒーレント・データを提供するステップと、

前記データベースが更新済みマッピングが保留中であるという表示を前記ノードに出力するステップと、

前記ノードが前記記憶装置へのデータ要求を中断するステップと、

前記ノードが未処理のデータ要求が完了するのを待つステップと、  
前記ノードが前記マッピングの内部表現を無効にするステップと、  
前記ノードが前記データベースに肯定応答信号を出力するステップと、  
前記データベースがアクティブ・ノードからの前記肯定応答信号を待つステップと、  
前記データベースが前記マッピングを更新するステップと、  
前記データベースが前記更新が完了したという表示を前記ノードに出力するステップと、  
前記ノードが前記データベースから前記マッピングの更新済みバージョンを要求するステップと、  
前記ノードが前記記憶装置への前記データ要求の送信を再開するステップとを含む方法。

【請求項59】 前記データベースがアクティブ・ノードから肯定応答信号を受け取らない場合、前記データベースが前記ノードに取消し表示を出力して、前記マッピングの前記内部表現を再活動化する、請求項58に記載の方法。

【請求項60】 前記ノードへの前記表示が前記データベースからの同期信号である、請求項58に記載の方法。

【請求項61】 前記同期信号が前記1つまたは複数のノードに同時に発行される、請求項60に記載の方法。

【請求項62】 データを失うかまたは破損することなしに前記マッピングが更新される、請求項61に記載の方法。

【請求項63】 記憶装置に対するノードのマッピングを更新する方法であって、

前記マッピングを高可用性データベースに記憶するステップであって、前記データベースが前記ノードによってアクセス可能であり、障害が存在する状態で前記ノードにコヒーレント・データを提供するステップと、

前記データベースが更新済みマッピングが保留中であるという表示を前記ノードに出力するステップと、

前記ノードが前記記憶装置へのデータ要求を中断するステップと

前記データベースが前記マッピングを更新するステップと、  
前記データベースが前記更新が完了したという表示を前記ノードに出力するステップと、  
前記ノードが前記記憶装置への前記データ要求の送信を再開するステップとを含む方法。

【請求項64】 前記データベースが前記マッピングを更新する前に、  
前記ノードが前記データベースに肯定応答信号を出力するステップと、  
前記データベースが前記アクティブ・ノードからの前記肯定応答信号を待つステップとをさらに含む、請求項63に記載の方法。

【請求項65】 前記ノードが前記データベースに肯定応答信号を出力する前に、  
前記ノードが未処理のデータ要求が完了するのを待つステップと、  
前記ノードが前記マッピングの内部表現を無効にするステップとをさらに含む、請求項64に記載の方法。

【請求項66】 前記ノードが前記記憶装置への前記データ要求の送信を再開する前に、前記ノードが前記データベースから前記マッピングの更新済みバージョンを要求するステップをさらに含む、請求項65に記載の方法。

【請求項67】 前記データベースがアクティブ・ノードから肯定応答信号を受け取らない場合、前記データベースが前記ノードに取消し表示を出力して、前記マッピングの前記内部表現を再活動化する、請求項63に記載の方法。

【請求項68】 前記ノードへの前記表示が前記データベースからの同期コマンドである、請求項67に記載の方法。

【請求項69】 前記同期信号が前記1つまたは複数のノードに同時に発行される、請求項68に記載の方法。

【請求項70】 データを失うかまたは破損することなしに前記マッピングが更新される、請求項69に記載の方法。

【請求項71】 記憶装置に対するノードのマッピングを更新するためのプログラム命令を含むコンピュータ可読記憶媒体であって、前記プログラム命令が分散コンピューティング・システムの複数のノード上で実行され、前記プログラ

ム命令が、

前記マッピングを高可用性データベースに記憶するステップであって、前記データベースが前記ノードによってアクセス可能であり、障害が存在する状態で前記ノードにコヒーレント・データを提供するステップと、

前記データベースが更新済みマッピングが保留中であるという表示を前記ノードに出力するステップと、

前記ノードが前記記憶装置へのデータ要求を中断するステップと、

前記ノードが未処理のデータ要求が完了するのを待つステップと、

前記ノードが前記マッピングの内部表現を無効にするステップと、

前記ノードが前記データベースに肯定応答信号を出力するステップと、

前記データベースがアクティブ・ノードからの前記肯定応答信号を待つステップと、

前記データベースが前記マッピングを更新するステップと、

前記データベースが前記更新が完了したという表示を前記ノードに出力するステップと、

前記ノードが前記データベースから前記マッピングの更新済みバージョンを要求するステップと、

前記ノードが前記記憶装置への前記データ要求の送信を再開するステップとを  
実施するように動作可能である、コンピュータ可読記憶媒体。

【請求項72】 第1のノードと、第2のノードと、第3のノードと、データ通信リンクとを含む分散コンピューティング・システムのデータ・トランスポート・システムであって、

前記分散コンピューティング・システムのアクティブ・ノードの数と、前記アクティブ・ノード間のリンクの数を決定し、前記リンクにより接続を確立するように構成された構成モジュールと、

前記構成モジュールから前記アクティブ・ノードの数と前記リンクの数を示すデータを受け取り、第1のアクティブ・ノードにデータを転送するための要求をクライアントから受け取り、1つまたは複数の前記リンクを介して前記第1のアクティブ・ノードに前記データを送るように構成された接続モジュールとを含み

前記アクティブ・ノードの数を変更されると、前記構成モジュールが前記変更を前記接続モジュールに通知し、前記接続モジュールは前記クライアントにとってトランスペアレントな前記アクティブノードへの前記接続を再確立するように構成される、データ・トランスポート・システム。

【請求項73】 前記構成モジュールが高可用性データベースから前記リンクの数を示すデータを受け取る、請求項72に記載のデータ・トランスポート・システム。

【請求項74】 前記高可用性データベースが前記ノードのすべてによってアクセス可能であり、各ノードが一貫したデータを受け取る、請求項73に記載のデータ・トランスポート・システム。

【請求項75】 ノード障害が存在する状態で前記高可用性データベースが一貫したデータを記憶する、請求項74に記載のデータ・トランスポート・システム。

【請求項76】 前記構成モジュールがデーモンである、請求項72に記載のデータ・トランスポート・システム。

【請求項77】 前記接続モジュールがカーネル・モジュールである、請求項76に記載のデータ・トランスポート・システム。

【請求項78】 前記構成モジュールと前記接続モジュールがプライベート・インタフェースを介して通信する、請求項72に記載のデータ・トランスポート・システム。

【請求項79】 前記データ通信リンクが、前記第1のノード上で実行される前記接続モジュールのインスタンスと前記第2のノード上で実行される前記接続モジュールのインスタンスとのインタフェースを提供する、請求項72に記載のデータ・トランスポート・システム。

【請求項80】 前記データ通信リンクが前記第1のノードと前記第2のノードとの間の複数の物理リンクを含み、前記構成モジュールが前記複数の物理リンクを1つの仮想リンクとして管理する、請求項79に記載のデータ・トランスポート・システム。

【請求項81】 前記データ・トランスポート・システムが複数のクライアントに対応する、請求項72に記載のデータ・トランスポート・システム。

【請求項82】 前記複数のクライアントが前記ノード間でメッセージを送受する、請求項81に記載のデータ・トランスポート・システム。

【請求項83】 前記構成モジュールがコールバック機能を介して他のアクティブ・ノードから受け取ったメッセージをクライアントに通知する、請求項82に記載のデータ・トランスポート・システム。

【請求項84】 前記データ・トランスポート・システムによって転送されたデータがメッセージを含む、請求項72に記載のデータ・トランスポート・システム。

【請求項85】 1つのメッセージが制御メッセージとデータ部分の両方を含む、請求項84に記載のデータ・トランスポート・システム。

【請求項86】 前記接続モジュールがメッセージ用の記憶空間を割り振り、解放する、請求項72に記載のデータ・トランスポート・システム。

【請求項87】 あるメッセージからのデータがもはや不要になったときにクライアントが前記接続モジュールに通知し、前記接続モジュールが前記メッセージに関連する記憶空間を解放する、請求項86に記載のデータ・トランスポート・システム。

【請求項88】 複数のノードと1つのデータ通信バスとを含む分散コンピューティング・システム内でデータを移送する方法であって、

前記分散コンピューティング・システム内の物理資源を決定するステップであって、前記物理資源が前記分散コンピューティング・システムのアクティブ・ノードと、前記アクティブ・ノード間のアクティブ・リンクとを含むステップと、

前記アクティブ・リンクにより接続を確立するステップと、

前記アクティブ・ノードのうちの第1のノードにデータを送るためのデータ・アクセス要求を受け取るステップと、

前記アクティブ・リンクのうちの1つまたは複数により前記第1のアクティブ・ノードに前記データを送るステップと、

前記物理資源が変更されたことを決定するステップと、

前記変更済み物理資源への接続を再確立するステップとを含み、  
前記変更済み資源の決定と前記リンクの再確立がクライアントにとってトランスペアレントなものである方法。

【請求項89】 アクティブ・ノード間の複数のリンクが1つの論理リンクとして扱われる、請求項88に記載の方法。

【請求項90】 前記物理資源の決定がデーモン・モジュールによって実行される、請求項88に記載の方法。

【請求項91】 前記アクティブ・リンクによる接続の確立がデーモン・モジュールによって実行される、請求項90に記載の方法。

【請求項92】 前記アクティブ・ノードへの前記データの運搬がカーネル・モジュールによって実行される、請求項91に記載の方法。

【請求項93】 複数のクライアントがサポートされ、前記アクティブ・ノードに運搬される前記データが、データ・アクセス要求を要求したクライアントの識別を含む、請求項88に記載の方法。

【請求項94】 前記運搬されたデータが制御部分とデータ部分とを含む、請求項88に記載の方法。

【請求項95】 アクティブ・ノードに運搬される前記データを記憶するためのメモリ空間を割り振るステップと、  
前記メモリ空間を解放するステップとをさらに含む、請求項88に記載の方法。

【請求項96】 前記クライアントに向けられるデータを受取りを宛先ノード側のクライアントに通知するステップをさらに含む、請求項89に記載の方法。

【請求項97】 物理資源の決定が、物理資源のリストを記憶する高可用性データベースにアクセスすることを含む、請求項89に記載の方法。

【請求項98】 前記高可用性データベースが前記アクティブ・ノードによってアクセス可能であり、前記アクティブ・ノードが一貫した構成データを有する、請求項97に記載の方法。

【請求項99】 複数のノードと1つのデータ通信リンクとを含む分散コン



ピューティング・システム内でデータを移送するためのプログラム命令を含むコンピュータ可読記憶媒体であって、前記プログラム命令が前記分散コンピューティング・システムの前記複数のノード上で実行され、前記プログラム命令が、

前記分散コンピューティング・システム内の物理資源を決定するステップであって、前記物理資源が前記分散コンピューティング・システムのアクティブ・ノードと、前記アクティブ・ノード間のアクティブ・リンクとを含むステップと、

前記アクティブ・リンクにより接続を確立するステップと、

前記アクティブ・ノードのうちの第1のノードにデータを送るためのデータ・アクセス要求を受け取るステップと、

前記アクティブ・リンクのうちの1つまたは複数により前記第1のアクティブ・ノードに前記データを送るステップと、

前記物理資源が変更されたことを決定するステップと、

前記変更済み物理資源への接続を再確立するステップとを実施するように動作可能であり、

前記変更済み資源の決定と前記接続の再確立がクライアントにとってトランスペアレントなものである、コンピュータ可読記憶媒体。

【請求項100】 アクティブ・ノード間の複数のリンクが1つの論理リンクとして扱われる、請求項99に記載のコンピュータ可読記憶媒体。

【請求項101】 前記物理資源の決定がデーモン・モジュールによって実行される、請求項99に記載のコンピュータ可読記憶媒体。

【請求項102】 前記アクティブ・リンクによる接続の確立がデーモン・モジュールによって実行される、請求項101に記載のコンピュータ可読記憶媒体。

【請求項103】 前記アクティブ・ノードへの前記データの運搬がカーネル・モジュールによって実行される、請求項102に記載のコンピュータ可読記憶媒体。

【請求項104】 アクティブ・ノードに運搬される前記データを記憶するためのメモリ空間を割り振るステップと、

前記メモリ空間を解放するステップとをさらに含む、請求項99に記載のコン

コンピュータ可読記憶媒体。

【請求項105】 前記クライアントに向けられるデータの受取りを宛先ノード側のクライアントに通知するステップをさらに含む、請求項99に記載のコンピュータ可読記憶媒体。

【請求項106】 物理資源の決定が、物理資源のリストを記憶する高可用性データベースにアクセスすることを含む、請求項100に記載のコンピュータ可読記憶媒体。

【請求項107】 前記高可用性データベースが前記アクティブ・ノードによってアクセス可能であり、前記アクティブ・ノードが一貫した構成データを有する、請求項106に記載のコンピュータ可読記憶媒体。

## 【発明の詳細な説明】

## 【0001】

## (発明の背景)

## (1. 発明の分野)

本発明は、分散コンピューティング・システムの分野に関し、より詳細には分散仮想記憶装置に関する。

## 【0002】

## (2. 関連技術の説明)

クラスタなどの分散コンピューティング・システムは2つまたはそれ以上のノードを含むことがあり、それらのノードはコンピューティング・タスクを実行するために使用される。一般的に言えば、ノードとは、1つまたは複数のコンピューティング・タスクを実行するように設計された1つの回路グループである。1つのノードは、1つまたは複数のプロセッサと、1つのメモリと、インタフェース回路とを含むことができる。一般的に言えば、クラスタとは、ノード間でデータを交換する能力を有する2つまたはそれ以上のノードからなるグループである。あるノードで特定のコンピューティング・タスクを実行することができるが、他のノードは無関係のコンピューティング・タスクを実行する。あるいは、特定のコンピューティング・タスクを全体として実行するのに必要な時間を削減するために、そのコンピューティング・タスクのコンポーネントをノード間に分散することができる。一般的に言えば、プロセッサとは、1つまたは複数のオペランドの演算を実行して結果を生成するように構成されたデバイスである。演算は、プロセッサによって実行される命令に応答して実行することができる。

## 【0003】

1つのクラスタ内のノードは、そのノードに結合された1つまたは複数の記憶装置を有することができる。一般的に言えば、記憶装置とは、大量のデータを格納することができる持続性装置である。たとえば、記憶装置は、ディスク装置などの磁気記憶装置またはコンパクト・ディスク装置などの光学記憶装置にすることができる。ディスク装置は記憶装置の一例に過ぎないが、「ディスク」という用語は本明細書全体を通して「記憶装置」と交換して使用することができる。記

記憶装置に物理的に接続されたノードは記憶装置に直接アクセスすることができる。記憶装置はあるクラスタの1つまたは複数のノードに物理的に接続することができるが、その記憶装置はあるクラスタのすべてのノードに物理的に接続できるわけではない。ある記憶装置に物理的に接続されていないノードはその記憶装置に直接アクセスすることはできない。クラスタによっては、記憶装置に物理的に接続されていないノードは、ノード同士を接続するデータ通信リンクを介してその記憶装置に間接的にアクセスすることができる。

#### 【0004】

あるノードがクラスタ内のどの記憶装置にもアクセスできる、すなわちそのノードに記憶装置が物理的に接続されているかのようにすることは有利である。たとえば、Oracle Parallel Serverなどの一部のアプリケーションでは、UNIXデバイス・セマンティクスにより1つのクラスタ内のすべての記憶装置にアクセスしなければならない場合がある。あるノードに物理的に接続されていないが、あるノードに物理的に接続されているように見える記憶装置は、仮想装置または仮想ディスクと呼ばれる。一般的に言えば、分散仮想ディスク・システムとは、2つまたはそれ以上のノード上で動作するソフトウェア・プログラムであり、クライアントと1つまたは複数の記憶装置とのインターフェースとなり、その1つまたは複数の記憶装置がそのノードに直接接続されているというように見えるソフトウェア・プログラムである。一般的に言えば、クライアントとは、あるアクションを開始するためにプログラムにアクセスするプログラムまたはサブルーチンである。クライアントは、アプリケーション・プログラムである場合もあれば、オペレーティング・システム・サブルーチンである場合もある。

#### 【0005】

残念ながら、従来の仮想ディスク・システムは、一貫した仮想ディスク・マッピングを保証していない。一般的に言えば、記憶装置マッピングは、ある記憶装置がどのノードに物理的に接続されるか、ならびにこれらのノード上のどのディスク装置がその記憶装置に対応するかを識別するものである。ある仮想装置のある記憶装置にマッピングするノードとディスク・デバイスは、ノード/ディスク

対と呼ぶ場合もある。仮想装置マッピングは、許可およびその他の情報を含むこともある。ノード障害などの障害が発生した場合にマッピングが持続性のものであることが望ましい。あるノードが他のノードの支援なしにある装置と通信できる場合、そのノードはその装置に物理的に接続される。

#### 【0006】

クラスタは、ボリューム・マネージャを実装することができる。ボリューム・マネージャとは、そのクラスタの記憶資源を管理するツールである。たとえば、ボリューム・マネージャは、2つの記憶装置をミラーして1つの高可用性ボリュームを作成することができる。他の実施形態では、ボリューム・マネージャは、複数の記憶装置にわたるファイルの部分を記憶するストライピングを実施することができる。従来の仮想ディスク・システムは、記憶装置の上下いずれかに層状に重ねたボリューム・マネージャをサポートすることができない。

#### 【0007】

他の望ましい特徴としては、ノード障害または記憶装置経路障害などの障害が存在する状態でデータ・アクセス要求が確実に実行されるような高可用性のデータ・アクセス要求を含む。一般的に言えば、記憶装置経路とは、あるノードからある記憶装置への直接接続部である。一般的に言えば、データ・アクセス要求とは、データを読み書きするための記憶装置への要求である。

#### 【0008】

仮想ディスク・システムでは、複数のノードが1つの記憶装置の多くの表現を持つことがある。残念ながら、従来のシステムは、各ノード上の表現が一貫した許可データを有することを保証する確実な手段を提供しない。一般的に言えば、許可データは、どのユーザが装置、ディレクトリ、またはファイルにアクセスするための許可を有するかを識別するものである。許可としては、読取り許可、書き込み許可、実行許可を含むことができる。

#### 【0009】

さらに、あるクラスタのノードを追加または除去する能力を有するか、またはそのクラスタが動作している間に既存のノードと記憶装置との接続を変更することが望ましい。この能力は、そのクラスタを低下させることができないクリティ

カル・アプリケーションで使用するクラスタでは特に重要である。この能力により、物理的資源（ノードと記憶装置など）をシステムに追加するか、あるいはクラスタ内のデータ・アクセス要求を損なわずに修理および交換を実施することができる。

#### 【0010】

##### （発明の概要）

上記で概要を示した問題は、本発明による高可用性仮想ディスク・システムによって大部分が解決される。一実施態様における高可用性仮想ディスク・システムは、各記憶装置とクラスタ内の各ノードとのインタフェースを提供する。ノードの見地からすると、各記憶装置がそのノードに物理的に接続されているとみることができる。あるノードがある記憶装置に物理的に接続されている場合、仮想ディスク・システムはその記憶装置に直接アクセスできる。あるいは、そのノードがある記憶装置に物理的に接続されていない場合、仮想ディスク・システムはそのクラスタ内にあってその記憶装置に物理的に接続されている他のノードを通じてその記憶装置にアクセスできる。一実施態様では、すべてのノードが1つのデータ通信リンクを介して通信する。ある記憶装置が直接アクセスされるかまたは他のノードを介してアクセスされるかは、その記憶装置にアクセスするクライアントにとってトランスペアレントなものである。

#### 【0011】

一実施態様では、ノードは仮想ディスクのマッピングを記憶装置に記憶する。たとえば、各アクティブ・ノードは、各仮想装置用の一次ノード／ディスク対と二次ノード／ディスク対を識別するマッピングを記憶することができる。各ノード／ディスク対は、その記憶装置に物理的に結合されたノードと、その記憶装置に対応するそのノード上のディスク装置を識別する。二次ノード／ディスク対は、代替ノード／ディスク対ともいう場合がある。そのノードが一次ノード／ディスク対を介してある記憶装置にアクセスできない場合、そのノードは二次ノード／ディスク対を介してデータ・アクセス要求を再試行することができる。障害が存在する状態でノード間の一貫したマッピングを維持するために、そのマッピングを高可用性データベースに記憶することができる。高可用性データベースは障

害が存在する状態でも1つの一貫したデータ・コピーを維持するので、高可用性データベースに照会する各ノードは、同じマッピングを取得することになる。また、高可用性データベースを使用して、仮想装置へのアクセスを制御するための許可データを記憶することもできる。高可用性データベースは障害が存在する状態でも1つの一貫した許可データ・コピーを維持するので、そのデータベースに照会する各ノードは、同じ許可データを取得することになる。

#### 【0012】

本発明による仮想ディスク・システムの特徴の1つは、システムの高い可用性である。一実施態様では、仮想ディスク・システムは、それが受け取ったすべてのデータ・アクセス要求を記憶し、エラーが発生した場合にその要求を再試行する。たとえば、データ・アクセス要求を開始し、要求側ノードと呼ばれるノードの仮想ディスク・システムは、すべての未処理のデータ要求を記憶することができる。宛先ノード、すなわち、そのデータ・アクセス要求が向けられるノードがそのデータ・アクセス要求を完了できない場合、要求側ノードに対してエラー表示を返すことができ、要求側ノードはその記憶装置に接続されている代替ノードにそのデータ・アクセス要求を再送することができる。このエラー検出および再試行は自動的に実行され、クライアントにとってトランスペアレントなものである。他の例では、ノード障害が発生した場合、仮想ディスク・システムは、アクティブ・ノードの変更済みリストを受け取り、その記憶装置に結合されたアクティブ・ノードに不完全なデータ・アクセス要求を再送することができる。この再構成および再試行もクライアントにとってトランスペアレントなものである。

#### 【0013】

本発明による仮想ディスク・システムの他の特徴は、クラスタが動作している間にそのクラスタを再構成できる能力である。あるクラスタを再構成すると、記憶装置に対する仮想ディスクのマッピングを更新することができる。エラーを防止するため、マッピングを更新する前にそのクラスタのすべてのノードに対して同期コマンドを実行または操作することができる。この同期コマンドにより、ノードはデータ・アクセス要求の発行を停止する。マッピングを更新した後、他の同期コマンドにより、そのノードはデータ・アクセス要求の発行を再開する。

## 【0014】

仮想ディスク・システムは、ボリューム・マネージャと記憶装置とのインタフェースまたはクライアントとボリューム・マネージャとのインタフェースとして機能するように設計することができる。前者の構成では、クライアントはボリューム・マネージャにインタフェースし、ボリューム・マネージャは仮想ディスク・システムにインタフェースする。後者の構成では、クライアントは仮想ディスク・システムにインタフェースし、仮想ディスク・システムはボリューム・マネージャにインタフェースする。

## 【0015】

本発明の他の目的および利点は、以下に示す詳細説明を読み、添付図面を参照すると明らかになるだろう。

## 【0016】

本発明は様々な変更および代替形式が可能であるが、その具体的な実施形態を例証として添付図面に示し、本明細書に詳細に説明する。しかし、図面およびそれに対する詳細な説明は開示した特定の形式に本発明を限定するためのものではなく、むしろ、その意図は特許請求の範囲によって定義した本発明の精神および範囲に含まれるすべての変更態様、同等態様、代替態様を含むことである。

## 【0017】

(発明の詳細な説明)

次に図1に移行すると、本発明の一実施形態によるクラスタ構成のブロック図が示されている。クラスタ100は、データ通信リンク102と、3つのノード104A～104Cと、3つの記憶装置108、110、112とを含む。データ通信リンク102は、ノード間でデータを転送するためのデータ通信経路である。データ通信リンク102は、マルチドロップ・リンクまたはポイントツーポイント・リンクを企図している。たとえば、データ通信リンク102は3つのポイントツーポイント・リンクを含むことができる。第1のリンクはノード104Aと104Bとの間の通信経路で、第2のリンクはノード104Aと104Cとの間の通信経路で、第3のリンクはノード104Bと104Cとの間の通信経路である。一実施形態のデータ通信リンク102はスケラブル・コヒーレント・



インタフェース (scalable coherent interface: S C I) を実装する。特定の  
実施形態のクラスタは、S C Iによりデータを転送するためにT C P / I Pプロ  
トコルを実装している。例示のためにのみ3つのノードを示していることに留意  
されたい。他の実施形態ではそれより多いかまたは少ないノードを使用すること  
もできる。

#### 【0018】

図示の実施形態では、記憶装置108がノード104Aに物理的に接続され、  
記憶装置110がノード104Bに物理的に接続され、記憶装置112がノード  
104Cに物理的に接続されている。記憶装置108～112は一般に、それが  
接続されているノードのメモリの記憶容量を上回る記憶容量を有する。データは  
、ノードによって現在使用されていない記憶装置108～112に記憶され、そ  
の記憶装置からのデータがそのデータが必要になったときにそのノードのメモリ  
に記憶またはキャッシュされる。図示の実施形態では、記憶装置が1つのノード  
のみに物理的に接続されている。代替実施形態では、1つの記憶装置を複数のノ  
ードに物理的に接続することができる。複数の物理接続により、ある記憶装置に  
物理的に接続された1つのノードが故障した場合または記憶装置経路が故障した  
場合でもその記憶装置にアクセスすることができる。

#### 【0019】

同じ分散プログラムの複数のインスタンスが各ノード上で動作することができ  
る。たとえば、ボリューム・マネージャ105Aとボリューム・マネージャ10  
5Bは同じ分散ボリューム・マネージャ・プログラムの異なるインスタンスであ  
る。これらのインスタンスは、データ通信リンク102を介して互いに通信する  
ことができる。各インスタンスには、同じ参照番号とそれに続く固有に英字、た  
とえば、105Aまたは105Bが付与される。簡潔にするため、分散プログラ  
ムは、まとめて参照番号のみを使用する、たとえば、ボリューム・マネージャ1  
05とすることができる。

#### 【0020】

ノード104Aは、ボリューム・マネージャ105Aと仮想ディスク・システ  
ム106Aを含む。図示の実施形態の仮想ディスク・システム106Aは、ボ

リユーム・マネージャ105と記憶装置108～112とのインタフェースとなっている。ボリューム・マネージャ105Aの見地からすると、各記憶装置はノード104Aに物理的に接続されているように見える。仮想ディスク・システム106は複数のノード上で動作する分散プログラムである。図示の実施形態では、仮想ディスク・システム106の1つのインスタンスが各ノードで動作している。仮想ディスク・システム106Aは、ノード104Aで動作する仮想ディスク・システム106のインスタンスであり、記憶装置108～112をそれぞれ表す3つの仮想装置（VD1、VD2、VD3）を含む。ボリューム・マネージャ105は、自身のノードに物理的に接続された記憶装置に伝達するのと同じように仮想装置に伝達する。一実施形態では、ボリューム・マネージャ105はUNIXデバイス・ドライバ・セマンティクスを使用する。記憶装置108（すなわち、VD1）へのデータ・アクセス要求は仮想ディスク・システム106Aから記憶装置108に直接運搬される。記憶装置110および112（すなわち、VD2およびVD3）へのデータ・アクセス要求はデータ通信リンク102によりこれらの装置に物理的に接続されたそれぞれのノードに運搬される。

#### 【0021】

各ノードの仮想ディスクは別個の装置であることに留意されたい。たとえば、ノード104A、104B、104CのVD1は、それぞれ固有のデバイス・ドライバによって管理される固有の装置である。装置は固有であるが、各VD1装置は同じ物理記憶装置にマッピングする。換言すれば、ノード104AのVD1にデータを書き込むことは、ノード104Bまたは104CのVD1にデータを書き込むのと同様に、記憶装置108にデータを記憶する。各記憶装置が複数のノードに物理的に接続できることにさらに留意されたい。この場合、その装置に物理的に接続された各ノードは、記憶装置にインタフェースする異なるデバイス・ドライバを有する。

#### 【0022】

図示の実施形態では、ボリューム・マネージャ105Aのボリューム1（V1）がVD1およびVD2に結合されている。一実施形態では、ボリューム・マネージャ105Aがこれらの装置をミラーすることもできる。代替実施形態では、

ボリューム・マネージャ105Aが他の仮想装置に結合された他のボリュームを含むこともできる。たとえば、第2のボリューム・マネージャ105AはVD2およびVD3に結合することができる。

#### 【0023】

ノード104Bおよび104Cでは、ボリューム・マネージャ(105Bおよび105C)および仮想ディスク・システム(106Bおよび106C)がボリューム・マネージャ105Aおよび仮想ディスク・システム106Aと実質的に同じように動作する。図示の実施形態では、ボリューム・マネージャ105Bのボリューム2(V2)が仮想ディスク・システム106BのVD2およびVD3に結合されている。仮想ディスク・システム106Bは、記憶装置110に直接アクセスし、通信インタフェース102およびノード104Cを介して記憶装置112にアクセスする。ボリューム・マネージャ105Cのボリューム3(V3)は仮想ディスク・システム106CのVD2およびVD3に結合されている。仮想ディスク・システム106Cは、記憶装置112に直接アクセスし、通信インタフェース102およびノード104Bを介して記憶装置110にアクセスする。

#### 【0024】

次に図2に移行すると、本発明の一実施形態による代替クラスタ構成のブロック図が示されている。クラスタ200は、データ通信リンク102と、3つのノード104A~104Cと、3つの記憶装置108、110、112とを含む。簡潔にするため、図1の構成要素と同様の構成要素には同じ参照番号が付与されている。図2においては、クライアントは、ボリューム・マネージャ105ではなく仮想ディスク・システム106にインタフェースする。仮想ディスク・システムがボリューム・マネージャにインタフェースし、ボリューム・マネージャは1つまたは複数の記憶装置にインタフェースする。この構成では、ボリューム・マネージャ105は仮想ディスク・システム106の下に層状に重ねられている。簡潔にするため、ノード104Aの動作についてのみ以下に説明する。ノード104Bおよび104Cは実質的に同じように動作する。

#### 【0025】

ノード104Aでは、クライアントは仮想ディスク・システム106Aにインタフェースする。クライアントの見地からすると、仮想ディスク・システム106Aは3つの別々の記憶装置として現れる。図2の3つの仮想装置は、ボリューム・マネージャが仮想ディスク・システムの下に層状に重ねられていることを反映するように仮想ボリューム(VV1、VV2、VV3)として表示されている。クライアントの見地からすると、仮想ボリュームは記憶装置のように動作する。たとえば、仮想ボリュームはUNIXデバイス・ドライバ・セマンティクスを使用することができる。クライアントは、仮想ディスク・システム106Aからクラスタの3つのボリュームのいずれにもアクセスすることができる。ボリューム・マネージャ105Aは記憶装置にインタフェースする。図示の実施形態では、ボリューム・マネージャ105Aのボリューム1(V1)が記憶装置108および110に結合されている。一実施形態では、ボリューム1は記憶装置108および110にデータをミラーすることができる。仮想ディスク・システム106Aの見地からすると、ボリューム・マネージャ105Aのボリューム1は記憶装置のように動作する。たとえば、そのボリュームはUNIXデバイス・ドライバのように動作することができる。

#### 【0026】

仮想ディスク・システム106Bの仮想ボリューム2(VV2)はボリューム・マネージャ105Bのボリューム2(V2)に直接インタフェースする。仮想ボリューム1および3は、データ通信リンク102を介してノード104Aのボリューム1およびノード105Cのボリューム3と通信する。同様に、仮想ディスク・システム106Cの仮想ボリューム3はボリューム・マネージャ105Cのボリューム3に直接インタフェースする。仮想ボリューム1および2は、データ通信リンク102を介してノード104Aのボリューム1およびノード105Bのボリューム2と通信する。図示の実施形態では、ボリューム・マネージャ105Bのボリューム2およびボリューム・マネージャ105Cのボリューム3はどちらも記憶装置110および112に物理的に接続されている。

#### 【0027】

ボリューム・マネージャと仮想ディスク・システムはどちらも記憶装置のよう

に動作するので、ボリューム・マネージャは仮想ディスク・システムの上または下のいずれかに層状に重ねることができる。したがって、それがボリューム・マネージャにインタフェースするかまたは仮想ディスク・システムにインタフェースするかはクライアントにとってトランスペアレントなものである。どちらの実施形態でも、クライアントは3つの信頼できる記憶装置に直接アクセスすることができるように見える。ボリューム・マネージャと仮想ディスク・システムはどちらも記憶装置に直接インタフェースすることができる。ボリューム・マネージャによっては、仮想ディスク装置の上に層状に重ねられたときにより良好に動作できるものもある。たとえば、ベリタスCVMなどのクラスタ・ボリューム・マネージャは仮想ディスク・システムの上に層状に重ねられたときに最も良好に動作するが、ソルスティス・ディスク・スイート(SDS)などの非分散ボリューム・マネージャは仮想ディスク・システムの下で動作しなければならない場合もある。ボリューム・マネージャは仮想ディスク・システムの下で動作するために分散しなければならないことに留意されたい。仮想ディスク・システムがそれらが1つの装置であるかのようにノードの仮想ディスクを管理するのと同様に、CVMなどの分散ボリューム・マネージャは、それらが1つのボリュームであるかのようにボリューム(V1、V2、V3)を管理できることにさらに留意されたい。

#### 【0028】

次に図3に移行すると、本発明の一実施形態によるクラスタの2つのノードで動作する仮想ディスク・システムのブロック図が示されている。図示の実施形態では、各ノードはユーザ部分とカーネルとを含む。ノード104Aのユーザ部分では、クラスタ・メンバシップ・モニタ(CMM)310Aと、クラスタ構成データベース(CCD)311Aと、クライアント312Aと、ネットディスク・デーモン(NDD)314Aと、クラスタ・トランスポート・インタフェース・デーモン(CTID)316Aとを含む。ノード104Aのカーネルは、ネットディスク・ドライバ(ND)318Aと、ネットディスク・マスタ(NM)320Aと、クラスタ・トランスポート・インタフェース(CTI)322A、クラスタ接続性モニタ(CCM)324Aと、ディスク・ドライバ326Aと、ネット

ワーク・トランスポート328Aとを含む。ノード104Bのユーザ部分は、クラスタ・メンバシップ・モニタ (CMM) 310Bと、クラスタ構成データベース (CCD) 311Bと、ネットディスク・デーモン (NDD) 314Bと、クラスタ・トランスポート・インタフェース・デーモン (CTID) 316Bとを含む。ノード104Bのカーネルは、ネットディスク・ドライバ (ND) 318Bと、ネットディスク・マスタ (NM) 320Bと、クラスタ・トランスポート・インタフェース (CTI) 322B、クラスタ接続性モニタ (CCM) 324Bと、ネットディスク・ドライバ326Bと、ネットワーク・トランスポート328Bとを含む。

#### 【0029】

図示の実施形態ではボリューム・マネージャが含まれていない。図1および図2に関連して前述したように、ボリューム・マネージャは仮想ディスク・システムの上または下のいずれかに実装することができる。ボリューム・マネージャが仮想ディスク・システムの上に実装される場合、クライアント312Aがボリューム・マネージャにインタフェースし、次にそのボリューム・マネージャがND318Aにインタフェースする。反面、ボリューム・マネージャが仮想ディスク・システムの下に実装される場合、NM320Aがボリューム・マネージャにインタフェースし、次にそのボリューム・マネージャがディスク・ドライバ326Aにインタフェースする。

#### 【0030】

CTID316Aという構成モジュールは、CTI322Aという接続モジュールを初期設定するデーモンである。クラスタの構成が変更されると、ノード316Aは初期設定される。CTID316AはCCD311Aに照会して構成情報を入手する。一実施形態の構成情報は、そのクラスタのノード間のリンクの数と、リンクに関連するプロトコルとを示す。一実施形態では、CTID316AがCMM310Aをさらに照会し、クラスタ内のアクティブ・ノードのリストなどのメンバシップ情報を入手する。CTID316Aは、ノード間のリンクにより接続を確立し、メンバシップ情報とリンク情報をCTI322Aに送る。CTID316Aは、プライベート相互接続によりCTI322Aに連絡することが

でき、入出力制御要求を使用することができる。

#### 【0031】

CCD311Aによって識別されるリンクは、物理リンクの場合もあれば、仮想リンクの場合もある。たとえば、CCM324Aは、CTI322Aによってアクセス可能な1つの仮想リンクとして一対の物理リンクを管理することができる。CCM324については図9に関連して以下に詳述する。

#### 【0032】

CCD311Aは、分散高可用性クラスタ・データベースのインスタンスの1つである。CCD311は障害が存在する状態でも一貫したデータを記憶する。CCD311にマッピング・データを記憶することにより、各ノードは障害が存在する状態でも同じマッピング情報を入手する。CCD311については、Slaughter他により1997年10月21日に出願され、「Highly available Distributed Cluster Configuration Database」という名称で本願譲受人に譲渡された同時係属特許出願第08/954796号に詳述されている。

#### 【0033】

CMM310は、クラスタ・メンバシップを監視する分散プログラムである。メンバシップが変更されると、CMM310はその変更を検出し、CTID316AおよびNDD314Aなどクラスタ内の他の資源に新しいメンバシップ情報を送る。メンバシップ変更の例としては、そのクラスタに加わるノードまたはそのクラスタを離れるノードを含む。一実施形態のCMM310は各構成に固有の構成番号を出力する。

#### 【0034】

NDD314Aは、新しい装置をオープンしたときまたは再構成中にND318Aを初期設定するデーモンである。再構成は、ノードがそのクラスタに加わったときまたはそのクラスタを離れるとき、あるいはノードが故障したときに行われる。一実施形態では、各仮想ディスク装置は個別に初期設定される。特定の一実施形態の仮想ディスク装置は、そのクラスタがその装置をオープンしたときにクラスタによって初期設定されるか、または再構成の前に仮想ディスク・ドライ

バをオープンした場合は再構成後にクラスタによって初期設定される。このため、すべての仮想ディスク装置がそれぞれの再構成後に初期設定されるわけではない。

#### 【0035】

一実施形態のND318Aは、オープンすべき装置のリストと、オープンした装置のリストを記憶する。クライアントがある装置をオープンするよう要求すると、ND318Aはオープンすべき装置のリストにその装置を追加する。NDD314Aはオープンすべき装置のリストに照会する。そのリストがオープンすべき装置を含む場合、NDD314AはCCD311Aに照会し、識別した装置に関するマッピング情報を入手する。NDD314Aは、CMM310Aにも照会して、アクティブ・ノードのリストなどのメンバシップ情報を入手することもできる。NDD314Aは、マッピング情報とメンバシップ情報をND318Aに送る。NDD314Aはプライベート相互接続によりND318Aに連絡することができ、入出力制御要求を使用することができる。

#### 【0036】

一実施形態では、ある装置に関するマッピング情報は、ある記憶装置に物理的に接続された一次および二次ノードと、その記憶装置に対応するこれらのノードのディスク装置とを識別する。ノードとディスクの各対はノード/ディスク対ともいう場合がある。一次および二次ノード/ディスク対とメンバシップ情報とに基づいて、ND318Aは、ある装置に関するデータ・アクセス要求を経路指定するためのノードを選択することができる。ND314AとCTI322Aが初期設定されると、仮想ディスク・システムはクライアント312Aからデータ・アクセス要求を受け入れる準備が整っている。

#### 【0037】

クライアント312Aは、それが記憶装置にアクセスするのと同じように仮想ディスク・システムの仮想装置にアクセスする。クライアントの見地からすると、各記憶装置またはボリュームはそのノードに物理的に接続されているように見える。図示の実施形態では、クライアント312Aがある記憶装置からのデータにアクセスする場合、クライアントはデータ・アクセス要求をND318Aに送



る。一実施形態では、クライアント312Aは、宛先記憶装置と、動作のタイプと、データを検索または記憶するための位置とをND312Aに対して指定する。残りの動作はクライアント312Aにとってトランスペアレントなものになる。ND318Aは、マッピングおよび現行メンバシップ情報に基づいて、どのノードにデータ・アクセス要求を送るかを決定する。一実施形態では、CCD311Aから入手したマッピング情報は、その記憶装置に物理的に接続された一次および二次ノードを含む。ND318Aは、一次ノードがアクティブである場合、そのデータ・アクセス要求を一次ノードに経路指定することができる。あるいは、一次ノードがアクティブではない場合、ND318Aはそのデータ・アクセス要求を二次ノードに経路指定する。その記憶装置にアクセスするためにどのノードを使用するかは、クライアント312Aにとってトランスペアレントなものになる。

#### 【0038】

ND318Aは、CTI322Aにデータ・アクセス要求を送り、どのノードにデータ・アクセス要求を送るかを指定する。CTI322Aがどのようにデータ・アクセス要求を宛先ノードに転送するかは、ND318Aおよびクライアント312Aにとってトランスペアレントなものになる。一実施形態では、その記憶装置がノード104Aに直接結合されている場合、ND318AはCTI322AではなくNM320Aにデータ・アクセス要求を送る。NM320Aはデータ・アクセス要求をディスク・ドライバ326Aに送り、次にそのディスク・ドライバはその記憶装置にアクセスする。一実施形態のNM320Aは、ND318Aのうち、ディスク・ドライバ326Aにインタフェースする部分である。ディスク・ドライバ326Aは、ノード104Aに物理的に接続された1つまたは複数の記憶装置にインタフェースする。

#### 【0039】

CTI322Aは複数のリンクを管理する。CTI322Aは分散プログラムCTI322のインスタンスの1つである。CTI322Aは、あるデータ・アクセス要求の宛先ノードへの1つまたは複数のリンクを管理することができる。たとえば、そのデータ・アクセス要求の宛先ノードがノード104Bである場合

、CTI322Aはそのノードへの3つのリンクを管理することができる。CTI322Aは、1つのリンクを介してノード104Bにすべてのデータを移送する場合もあれば、3つのリンクによりデータを分散する場合もある。CTI322Aは、宛先ノードで宛先クライアントを識別するためのフィールドをデータ・アクセス要求に付加することができる。ノード104BのCTI322Bは複数のクライアントに対応することができる。CTI322Aによってメッセージに付加されたフィールドは、CTI322Bがどのクライアントにそのデータを経路指定すべきかを識別するものである。たとえば、CTI322Aは、宛先クライアントをND318Bとして指定するデータを、ND318Aが受け取るデータ要求に付加することができる。

#### 【0040】

一実施形態のCCM324Aは、2つまたはそれ以上の冗長物理リンクを管理する。CTI322Aの見地からすると、冗長物理リンクは1つの論理リンクとして現れる。CCM324Aは、物理リンクによりCCM324Bとメッセージを交換する。CCM324の2つのインスタンスは、冗長リンクのうちのどちらが動作可能であるかに関して合意に達する。CMM324は、データを転送するために1つの動作可能な物理リンクを選ぶことができる。そのリンクが故障した場合、CCM324は、その故障を検出し、代替リンク上でデータを転送することができる。CTI322の見地からすると、各論理リンクは1つの高可用性リンクとして現れる。一実施形態のCCM324Aはそのクラスタ内の各ノードへのリンクを管理する。たとえば、CMM324Aはノード104Bおよび104Cへのリンクを管理することができる。

#### 【0041】

ネットワーク・トランスポート328Aは、データ通信リンク112のリンクによりプロトコル機能を実行する。一実施形態では、データ通信リンク112によりTCP/IPプロトコルを使用する。他の実施形態では、他のプロトコルを実装することができる。たとえば、低待ち時間接続性層（LLCL）、メッセージ受渡しインタフェース（MPI）、低オーバーヘッド通信（LOCO）などの高速プロトコルを使用することができる。

## 【0042】

ノード104Bでは、ネットワーク・トランスポート328Bがデータ・アクセス要求を受け取り、適切なプロトコルを使用してデータをCTI322Bに移送する。CTI322Bは、データ・アクセス要求を部分的にデコードし、その宛先クライアントを決定することができる。図示の実施形態では、データはND318Bに経路指定される。ND318Bは、データ・アクセス要求を部分的にデコードし、宛先記憶装置を決定することができる。その記憶装置がノード104Bに物理的に結合されている場合、ND318Bは要求をNM320Bに送り、NM320Bはその要求をディスク・ドライバ326Bに送る。ディスク・ドライバ326Bはその記憶装置にアクセスする。データ・アクセス要求が読取りトランザクションである場合、要求されたデータはND318、CTI322、データ通信リンク112を介してクライアント312Aに戻される。

## 【0043】

本発明の一実施形態による仮想ディスク・システムの特徴の1つは高い可用性である。この仮想ディスク・システムは、ノード障害などの障害が存在する状態でデータ・アクセス要求が確実に実行されるように設計されている。この目的に向かって、ND318Aは保留データ・アクセス要求のリストを記憶する。データ・アクセス要求が正常に完了しない場合、仮想ディスク・システムは他のノードへのデータ・アクセス要求を再試行する。要求側ノードは、否定応答信号を受け取ることによって不完全なデータ・アクセス要求を検出する場合もあれば、宛先ノードがアクティブではないことを示す再構成データを受け取る場合もある。データ・アクセス要求が正常に完了した場合、それは保留データ・アクセス要求のリストから除去される。

## 【0044】

たとえば、ノード104Bがある記憶装置の一次ノードであり、ノード104Cがその記憶装置の二次ノードである場合が考えられる。ND318Aがその記憶装置にデータ・アクセス要求を送ると、それはそのデータ・アクセス要求を一次ノードに送ることができ、そのノードはノード104Bである。ノード104Bがデータ・アクセス要求を正常に完了できない場合、たとえば、ディスク・ド

ライバ326Bと記憶装置との間の記憶装置経路が機能しない場合、ノード104Aは、そのデータ・アクセス要求が正常に完了しなかったことを示す否定応答信号を受け取ることができる。次にノード104Aはデータ・アクセス要求を二次ノードに再送することができる。そのノードはノード104Cである。ノード104Aは、ノード104Bがその記憶装置と通信できないことを示す情報を記憶し、その後、新しいデータ・アクセス要求を他のノードに送ることができる。

#### 【0045】

代替例では、ノード104Bを動作不能にすることができる。一実施形態では、ノード104AがCMM310Aから取得したクラスタ・メンバシップ・データは、そのノードが動作不能であることを示す場合もある。したがって、ND318Aは、データ・アクセス要求を二次ノードに経路指定することができる。上記のように、障害が存在する状態でもデータ・アクセス要求が正常に完了する。

#### 【0046】

次に図4に移行すると、本発明の一実施形態によるネットディスク・ドライバの初期設定を示すブロック図が示されている。図4はノード104AにおけるND318Aの初期設定を示している。そのクラスタ内の他のネットディスク・ドライバの初期設定は実質的に同じように実行することができる。

#### 【0047】

一実施形態では、記憶装置にアクセスする前にその記憶装置をオープンする。たとえば、記憶装置を初期設定させるオープン・コマンドを実行することができる。同様に、各ノードの各仮想装置はそれにアクセスする前にオープンすることができる。クライアント312Aは、ND318Aにコマンドを出力して仮想装置をオープンする。ND318Aはオープンすべき装置をリストに記憶する。一実施形態のNDD314Aは、定期的にそのリストに照会して、どの装置を初期設定するかを決定する。代替実施形態のND318Aは、装置を初期設定する必要があることを示す信号をNDD314Aに出力することができる。NDD314Aは、オープンすべき装置に関するマッピング情報を入手するためにCCD311Aに照会し、現行メンバシップ情報についてはCMM310Aに照会する。NDD314Aはマッピングおよびメンバシップ情報をND318Aに送る。N

D318Aはマッピングおよびメンバシップ情報を構成ファイルに記憶する。ND318Aは、構成ファイルに記憶したマッピングおよびメンバシップ・データを使用して、ノードへのデータ・アクセス要求の経路指定を決定する。次にND318Aは、その装置がオープンされたことをクライアント312Aに通知する。

#### 【0048】

一実施形態では、各仮想装置に関するマッピング情報としては、仮想装置の名前と、一次ノードと、その一次ノードの記憶装置の名前（すなわち、その記憶装置に対応する装置の名前）と、二次ノードと、その二次ノードの記憶装置の名前とを含む。そのうえ、マッピング情報は、仮想装置の識別番号と、記憶装置のクラスタ特有の名前とを含むこともできる。

#### 【0049】

そのうえ、ND318Aは、マッピングおよびメンバシップ・データに関連する再構成番号を記憶する。再構成番号はCMM310Aから入手される。ND318Aは再構成番号を使用して、その現行メンバシップ・データが最近の構成に関して最新のものであるかどうかを判定する。

#### 【0050】

一実施形態では、クラスタの構成が変更されると、CMM310Aは、新しいメンバシップ情報をNDD314Aに通知する。たとえば、ノード障害が検出された場合、CMM314Aは、再構成が行われたことをNDD314Aに通知し、新しいメンバシップ・データをNDD314Aに送る。NDD314Aは新しいメンバシップ情報をND318Aに送り、そのND318Aはマッピング情報とともに新しいメンバシップ情報を使用して将来のデータ・アクセス要求を経路指定する。

#### 【0051】

一実施形態では、ファイルシステムがノードの仮想ディスクを管理する。このファイルシステムはネットディスク・ファイルシステム（NDFS）と呼ぶこともできる。NDFSはあるノードが仮想ディスクをオープンしたときにその仮想ディスク用の特殊装置ファイルを作成するように構成されている。この特殊装置

ファイルは、オペレーティング・システム内で仮想ディスクを表すものである。

#### 【0052】

UNIXオペレーティング・システムなどのオペレーティング・システムでは、装置をファイルとして扱うことができる。ある装置に関連するファイル（装置ファイルまたは特殊装置ファイルという）は、通常、オペレーティング・システムのブートアップ・フェーズ中に実行される初期設定プログラムによって作成される。初期設定プログラムは、コンピュータ・システムに接続された物理装置を決定し、その物理装置に対応する装置ファイルを作成する。一実施形態では、ブートアップ中ではなく、最初にアクセスされたときに仮想装置が初期設定される。この状況ならびにその仮想ディスクをノードに物理的に接続できないことは、初期設定中に仮想ディスク用の装置ファイルを作成できないことを意味する。好ましくは仮想ディスクは他の装置のようにアクセス可能なので、NDFSは、最初にオープンされたときに仮想装置の装置ファイルを作成するように構成されている。一実施形態では、あるノードがある仮想装置を最初にオープンしたときのみ、装置ファイルが作成される。その後、その仮想装置をオープンしても、装置ファイルは作成されない。

#### 【0053】

一実施形態のNDFSは仮想装置をオープンするコマンドを検出する。これがその仮想装置がオープンされた最初の場合であれば、NDFSは作成要求をND318Aに送る。一実施形態のNDFSは、ND318Aへのプライベート・インタフェースを有する。ND318Aはリストとして作成するためにその仮想装置を記憶する。このリストは、オープンすべき装置を記憶するために使用するのと同じリストである場合もあれば、作成すべき装置用の個別のリストである場合もある。NDD314Aが定期的にそのリストに照会してどの装置を作成すべきかを決定する場合もあれば、ある装置を作成する必要があることを示す信号をND318AがNDD314Aに出力する場合もある。NDD314AはCCD311Aに照会し、オープンすべき装置に関する許可データを入手する。NDD314AはND318Aに許可データを送り、次にそのND318Aが許可データをNDFSに送る。NDFSは、CCD311Aから受け取った許可データによ

ってその装置に関する装置ファイルを作成することになる。一実施形態では、前述のように通常の装置オープン手順を使用して装置ファイルが作成されたあとで装置がオープンされる。その後、同じノードによって同じ装置をオープンすると、NDFSがかかわる必要なしに通常のオープン動作が行われる。したがって、性能上のハンディは装置を最初にオープンしたときにだけ発生する。その装置をオープンするためのその後のコマンドは、他のどの装置のオープンとも同じように実行される。

#### 【0054】

次に図5に移行すると、本発明の一実施形態によるクラスタ・トランスポート・インタフェースの初期設定を示すブロック図が示されている。図5はノード104AにおけるCTI316Aの初期設定を示している。そのクラスタ内の他のクラスタ・トランスポート・インタフェースの初期設定は実質的に同じように実行することができる。

#### 【0055】

一実施形態では、データ通信リンク102によりデータを転送する前に、CTID316Aは使用可能なリンクにより接続を確立する。初期設定中にCTID316Aは、現行クラスタ・メンバシップを識別するデータを求めてCMM310Aに照会し、どのリンクがどのノードに接続されるかを識別するデータを求めてCCD311Aに照会する。一実施形態のCCD311Aは、リンクの転送プロトコルなど、そのリンクに関する追加情報を記憶する。CTID316Aは、使用可能なリンクにより接続を確立し、リンク情報およびメンバシップ・データをCTI322Aに渡す。一実施形態のCTID316Aは使用可能なリンクによりTCP/IP接続を確立する。

#### 【0056】

CTI322Aはネットワーク・トランスポート328Aにインタフェースし、CTI322の他のインスタンスへのデータを交換する。一実施形態のネットワーク・トランスポート328AはCCM324Aにインタフェースし、それが1つまたは複数の冗長リンクを管理する。CTI322Aは、特定のノード宛てのデータ・アクセス要求を受け取ると、どの接続が要求側ノードを宛先ノードに

接続するかを決定する。CTI322Aは、どの接続（複数も可）上で宛先ノードにデータを移送するかを決定する。たとえば、CTI322Aがノード104Bへの3つのリンクによる接続を管理し、それがそのノード宛てのデータ・アクセス要求を受け取る場合、CTI322Aは1つの接続を介してすべてのデータを転送する場合もあれば、3つの接続のそれぞれによりデータの一部分を転送する場合もある。

#### 【0057】

クラスタが再構成されると、CMM310Aはその事象をCTID316Aに通知する。CTID316AはCCD311Aから新しいメンバシップ・データを入手し、その新しいメンバシップ・データと新しい構成番号をCTI322Aに送る。そのうえ、CTID316AはCCD311Aからリンク・データを入手することができ、そのデータをCTI322Aに送る。CTID322Aは、再構成が行われたときに接続を変更することができる。たとえば、CTID322Aは、クラスタ内の新しいノードに対してリンクにより接続を確立する場合もあれば、そのクラスタを離れるノードへの接続を放棄する場合もある。

#### 【0058】

次に図6に移行すると、本発明の一実施形態による仮想ディスク・システムの動作を示す流れ図が示されている。ステップ612では、ネットディスク・ドライバを初期設定する。ネットディスク・ドライバの初期設定については図7に関連して詳述する。ステップ614では、クラスタ・トランスポート・ネットワークを初期設定する。クラスタ・トランスポート・インタフェースの初期設定については図8に関連して詳述する。ステップ616では、ネットディスク・ドライバがクライアントからデータ・アクセス要求を受け取る。ステップ617では、ネットディスク・ドライバは、データ・アクセス要求と、それが正常に完了していない場合にデータ・アクセス要求を再発行するために必要な他のデータを記憶する。

#### 【0059】

ステップ618では、データ・アクセス要求を受け取るネットディスク・ドライバは、宛先装置が要求側ノードに物理的に接続されているかどうかを判定する



。宛先装置が要求側ノードに物理的に接続されている場合、ネットディスク・ドライバはステップ620で記憶装置上でデータ・アクセス要求を実行する。あるいは、記憶装置が要求側ノードに物理的に接続されていない場合、ネットディスク・ドライバはステップ620でデータ・アクセス要求を送るべきノードを検出する。一実施形態のネットディスク・ドライバは、各記憶装置ごとに一次および二次ノードを識別するマッピング情報を記憶する。特定の二実施形態では、ネットディスク・ドライバは、メンバシップ・データおよび/または前の不成功に終わったデータ・アクセス要求に基づいて、一次または二次ノードを選択する。ステップ624では、ネットディスク・ドライバは、クラスタ・トランスポート・インタフェースを介して選択した宛先ノードにデータ・アクセス要求を送る。

#### 【0060】

ステップ626では、クラスタ・トランスポート・インタフェースは、ネットディスク・ドライバにより宛先ノードにデータを転送するために1つまたは複数の接続を選択する。ステップ628では、クラスタ・トランスポート・インタフェースは、選択した接続（複数も可）を介して宛先ノードにデータ・アクセス要求を送る。ステップ630では、宛先ノード側のクラスタ・トランスポート・インタフェースは、データ・アクセス要求を受け取って宛先クライアントを決定するが、その宛先クライアントはこの例ではネットディスク・ドライバ、またはより詳細にはネットディスク・マスタである。ステップ632では、ネットディスク・マスタがデータ・アクセス要求を受け取り、宛先記憶装置にアクセスする。ステップ634では、宛先ノードのクラスタ・トランスポート・インタフェースが肯定応答または否定応答信号を要求側ノードに返す。データ・アクセス要求が読取り要求である場合、要求されたデータも要求側ノードに返されるであろう。

#### 【0061】

データ・アクセス要求の転送と並行して、ステップ638では、要求側ノードが宛先ノードからの状況信号を待つ。この状況信号は肯定応答または否定応答信号の形をとることができる。ステップ640では、肯定応答が受け取られたかどうかを判定する。肯定応答信号が受け取られた場合、動作はステップ616に継続する。あるいは、否定応答信号が受け取られた場合、ステップ642では、デ

ータ・アクセス要求を送るための代替ノードを選択し、動作はステップ624に継続する。

#### 【0062】

次に図7に移行すると、本発明の一実施形態によるネットディスク・ドライバの初期設定を示す流れ図が示されている。ステップ712では、ネットディスク・デーモンは、オープンすべき装置を求めてネットディスク・ドライバに照会する。判断ステップ714では、オープンする必要がある装置があるかどうかを判定する。どの装置もオープンする必要がある場合、実行はステップ712に継続する。あるいは、ネットディスク・デーモンがオープンすべき装置を検出した場合、ステップ716でネットディスク・デーモンがマッピング・データを求めてクラスタ構成データベースに照会する。このマッピング・データは仮想装置にマッピングされたノード／ディスク対を識別することができる。ステップ718では、ネットディスク・デーモンはメンバシップ・データを求めてクラスタ・メンバシップ・モニタに照会する。

#### 【0063】

ステップ720では、ネットディスク・デーモンはマッピングおよびメンバシップ・データをネットディスク・ドライバに送る。ステップ722では、ネットディスク・ドライバは、その装置に関するマッピング情報を更新し、そのメンバシップ情報を更新し、再構成番号を記録する。ステップ724では、ネットディスク・ドライバは、要求された装置がオープンされていることをクライアントに通知する。

#### 【0064】

次に図8に移行すると、本発明の一実施形態によるクラスタ・トランスポート・インタフェースの初期設定を示す流れ図が示されている。ステップ812では、クラスタ・トランスポート・インタフェース・デーモンは構成変更の表示を受け取る。あるいは、クラスタ・トランスポート・デーモンは、システム初期設定の表示を受け取る場合もある。ステップ814では、クラスタ・トランスポート・インタフェース・デーモンは、リンク情報を求めてクラスタ構成データベースに照会する。一実施形態では、リンク情報は、あるクラスタ内のノード間のリン

クの数と、どのリンクがどのノードに結合されているかということ、そのリンクが使用するプロトコルなどの情報とを含むことができる。ステップ816では、クラスタ・トランスポート・インタフェース・デーモンはメンバシップ情報を求めてクラスタ・メンバシップ・モニタに照会する。

#### 【0065】

ステップ818では、クラスタ・トランスポート・インタフェースがリンクにより接続を確立する。ステップ820では、クラスタ・トランスポート・インタフェース・デーモンがクラスタ・トランスポート・インタフェースにリンクおよびメンバシップ情報を送る。その場合、クラスタ・トランスポート・インタフェースはデータ・アクセス要求またはその他のメッセージを受け入れる準備が整っている。

#### 【0066】

次に図9に移行すると、本発明の一実施形態によるクラスタ・トランスポート・インタフェースのブロック図が示されている。クラスタ・トランスポート・インタフェースはデータ・トランスポート・システムの一例である。図9は、クラスタ・トランスポート・インタフェースの3つのインスタンス(322A~322C)と、3つのTCP/IPインタフェース(912A~912C)と、8つのクラスタ接続モニタ(914A~914H)とを含む。CTI322は、ノード間でメッセージを受け渡すための機能を備えた分散ソフトウェア・プログラムである。そのメッセージとしては、制御メッセージとデータ・ブロックを含むことができる。

#### 【0067】

クラスタ・トランスポート・インタフェース322のインスタンスは、クライアント・プログラム間でデータを受け渡す。たとえば、CTI322Aは、CTI322Aにとってクライアントであるネットディスク・ドライバからメッセージを受け取ることができる。一実施形態のメッセージは、その宛先ノードと、そのノードのディスク装置を指定するものである。CTI322Aは、どのリンクが宛先ノードに接続されるかを決定し、そのリンクのうちの1つによりメッセージを送る。宛先ノード側のクラスタ・トランスポート・インタフェースは、デー

タ・アクセス要求を受け取り、宛先クライアントを決定し、宛先クライアントにデータを送る。たとえば、CTI322Aは、ノード104A内のネットディスク・ドライバからノード104B内のネットディスク・ドライバにデータ・アクセス要求を経路指定することができる。CTI322Bは、データ・アクセス要求を受け取り、宛先クライアントを決定し、ノード104B内のネットディスク・ドライバにデータ・アクセス要求を送る。クライアントの見地からすると、CTI322Aは宛先ノードへの仮想リンクの1つとして現れる。

#### 【0068】

図示の実施形態では、CTI322は、他のノードにデータを転送するためにTCP/IPを使用する。CTID316Aは、初期設定中に各リンクによりTCP/IP接続を自動的に確立する。CTI322は、CCM914の適切なインスタンスにメッセージを転送するTCP/IP912Aにメッセージを送る。しかし、CTI322Aは、特定のデータ転送プロトコルに依存していない。TCP/IP912および/またはCCM914を変更することにより、CTI322はどのようなデータ・トランスポート・インタフェースまたは転送プロトコルにもインタフェースすることができる。

#### 【0069】

一実施形態のCTI322Aは、他のノードから受け取ったメッセージおよびデータを記憶するためのメモリを割り振り、クライアントがもはやそのデータを必要としなくなったときにそのメモリの割振りを解除する。一実施形態のCTI322は、コールバック機能を使用して、そのデータを受け取ったことをクライアントに示す。たとえば、CTI322Aはノード104Bに読取り要求を送ることができる。CTI322Aは、要求されたデータを受け取ると、要求側クライアントへのコールバック機能を使用して、そのデータが使用可能であることを示す。

#### 【0070】

クラスタ接続モニタ(CCM)914は、2つまたはそれ以上の物理リンクを1つの論理リンクとして管理する。図示の実施形態では、CCM914の一对のインスタンスが2つのリンクを管理する。代替実施形態では、CCM914の一

対のインスタンスがそれ以上のリンクを管理することができる。複数対の物理リンクはそのクラスタ内のあるノードを他のノードに接続する。たとえば、リンク916Aはノード104Aをノード104Bに結合し、リンク916Bはノード104Aをノード104Cに結合する。一実施形態では、CMM914によってリンクが冗長リンクとして扱われる。データは一方のリンクの障害が検出されるまでそのリンク上で転送され、その後、データはもう一方のリンク上で転送される。

#### 【0071】

CCM914は、どのリンクが動作可能であるかを決定し、両方の物理リンクにより、ときにはハートビート・メッセージと呼ばれるメッセージを交換することで障害を検出する。たとえば、CCM914AとCCM914Eは、ハートビート・メッセージを交換して、物理リンク916Aが動作可能であるかどうかを判定する。CCM914の2つのインスタンスは物理リンクのうちの一方を一次リンクとして選択する。一次リンクが故障した場合、CCM916はその障害を検出し、もう一方の物理リンク上でデータの転送を開始する。特定の一実施形態では、CCM916は、物理リンクを越えて不信頼データ・プロトコル(UDP)メッセージを交換して、そのリンクが動作可能であるかどうかを判定する。

#### 【0072】

CTI322の見地からすると、CCM914によって管理される各対の物理リンクは1つの論理リンクとして現れる。したがって、CTI322Aによって転送されるデータは、CTI322Aにとってトランスペアレントな2つの物理リンクのうちの一方で転送することができる。

#### 【0073】

図示の実施形態では、3つの論理リンク(916B~916D)がノード104Aをノード104Cに接続する。CTI322Aは、3つのリンクのうちのどのリンク上でデータを転送するかを決定する。一実施形態のCTI322Aは1つの論理リンク上ですべてのデータを転送することができる。代替実施形態のCTI322Aは各論理リンク上でデータの一部分を転送することができる。上記のように、どの論理リンク上またはいくつの論理リンク上でデータを転送するか

はクライアントにとってトランスペアレントなものである。

【0074】

次に図10に移行すると、本発明の一実施形態による装置許可を示す図が示されている。許可データはディレクトリのリストに関連して示されている。同様のリストは、生の仮想ディスク装置をリストするディレクトリで「ls-l」コマンドを実行することによって入手することができる。装置許可は装置そのものに関するものであって、その装置上のファイルまたはディレクトリに関するものではないことに留意されたい。生の装置（すなわち、その上にファイルシステムまたはファイルが一切置かれていない装置）は許可目的のためのファイルとして扱われる。

【0075】

フィールド1012は10個の文字を含む。第1の文字は、ディレクトリを識別する「d」または装置を識別する「-」のいずれかである。次の9つの文字は、3文字ずつ3つのグループである。各グループは、所有者、グループ、その他のための許可モードをそれぞれ表している。許可モードとしては、読取り（r）、書込み（w）、実行（x）を含む。各グループ内の1つの文字は各許可モードを表す。許可モードを表す英字が存在する場合、関連ユーザはその許可を得ている。あるいは、「-」が存在する場合、関連ユーザはその許可を得ていない。たとえば、あるユーザが「rwx」という許可を得ている場合、そのユーザは、読取り、書込み、実行の各許可を得ていることになる。あるいは、ユーザが「r--」という許可を得ている場合、そのユーザは読取り許可を得ているが、書込みまたは実行の各許可を得ていないことになる。第1のグループの3つの文字はその装置の所有者ための許可を表している。第2のグループの3つの文字はその装置に関連するグループのための許可を表している。最後のグループの3つの文字は他のユーザのための許可を表している。所有者とグループについては以下に詳述する。たとえば、フィールド1012内の許可が「drwx--x-x-」である場合、そのフィールドはディレクトリを表し、所有者が読取り、書込み、実行の各許可を得ており、グループとその他が実行許可のみを得ていることになる。

## 【0076】

フィールド1016はその項目の所有者を識別する。所有者はその装置を作成したユーザである。フィールド1018は関連ユーザのグループを識別する。グループはオペレーティング・システム内で定義される。フィールド1018は定義済みグループの1つを装置に関連付けるものである。他のユーザは所有者ではなく、選択したグループ内にも含まれない。前述のように、所有者、グループ、その他のユーザのために、それぞれ異なる許可を定義することができる。

## 【0077】

フィールド1022はその装置の最後の変更の日付と時刻を識別する。最後の変更が現行暦年の範囲内である場合、月、日、時刻が指定される。あるいは、最後の変更が現行暦年の範囲内ではない場合、月、日、年が指定される。フィールド1024は装置の名前を識別する。

## 【0078】

クラスタのノード間で一貫した許可データを保証するため、許可データは高可用性データベースに記憶することができる。一実施形態では、あるクラスタ内の複数のノードがある装置の表現を有する。障害が存在する状態でもノード間で一貫した許可データを維持するため、許可データはクラスタ構成データベース（CCD）に記憶される。

## 【0079】

一実施形態では、あるノードが最初に仮想装置をオープンすると、その装置のための許可データがCCDから読み取られ、その許可データによって装置ファイルが作成される。一実施形態の装置ファイルは、仮想装置があるノードによって最初にオープンされたときにのみ作成される。一実施形態では、各ノード上で動作するファイルシステムは、その装置の許可データを求めてCCDに照会するデーモンを含む。このデーモンは許可データをファイルシステムに返し、そのファイルシステムがその許可によって特殊装置ファイルを作成する。CCDはそのクラスタのどのノードでも照会することができ、障害が存在する状態でも一貫した情報を返すので、すべてのノードが一貫した許可データを有することになる。

## 【0080】

次に図11に移行すると、本発明の一実施形態による一貫した許可データの記憶およびアクセスを示す流れ図が示されている。ステップ1112では、許可データを高可用性データベースに記憶する。一実施形態の許可データは、装置許可と、装置の所有者と、装置に関連するグループとを含む。ステップ1114では、第1のノードは第1のノードの装置をオープンし、高可用性データベースからの許可データにアクセスする。ステップ1115では、そのノードは、許可データを含む、その装置に関する特殊装置ファイルをオープンする。ステップ1116では、第2のノードは、第2のノード上にあつて同じ物理装置に対応する装置をオープンし、許可データにアクセスする。ステップ1117では、そのノードは、第2のノードの許可データを含む、その装置に関する特殊装置ファイルをオープンする。高可用性データベースは一貫したデータを返すので、ノードは一貫した許可データを受け取る。

#### 【0081】

次に図12に移行すると、本発明の一実施形態による構成マッピングの更新を示す流れ図が示されている。ステップ1212では、更新が保留中であるという表示をノードに提供する。ステップ1214では、ノードは記憶装置へのデータ・アクセス要求を中断する。ステップ1216では、ノードは未処理のデータ・アクセス要求が完了するのを待つ。ステップ1218では、ノードは記憶装置に対する仮想ディスクのマッピングの内部表現を無効にする。ステップ1220では、ノードは、内部マッピング表現が無効になり、データ・アクセス要求が中断され、未処理のデータ・アクセス要求が完了したことを示す肯定応答信号を出力する。ステップ1222では、システムはすべてのアクティブ・ノードからの肯定応答信号を待つ。ステップ1224では、システムはそのマッピングを更新する。ステップ1226では、システムは、その更新が完了したという表示を出力する。ステップ1228では、ノードはマッピングの更新済みバージョンを要求する。ステップ1230では、ノードは、記憶装置へのデータ・アクセス要求の送信を再開する。

#### 【0082】

一実施形態での更新手順は、クラスタ構成データベース（CCD）によって調



整される。エラーを防止するため、マッピングはすべてのノード間で一貫して更新しなければならない。CCDは、保留中の更新をノードに通知し、その更新が完了したことを同期コマンドによりノードに通知する。一実施形態の同期コマンドは、CCD内の行が変更されるたびに必ず実行される。CCD内の行の変更中に実行すべきコマンドは、CCDに記憶されたデータに関連するフォーマット行で指定することができる。同期コマンドは、そのクラスタのすべてのノード上で並行して実行することができる。一実施形態のネットディスク同期コマンドは、ネットディスク・マッピングが変更されたときに実行される。ネットディスク同期コマンドの異なる呼出しは、変更のタイプに応じて実行することができる。CCDは、マッピングを変更する前に第1の同期コマンドを出力する。第2の同期コマンドは、データベースを更新したあとで実行することができる。

#### 【0083】

一実施形態では、すべてのノードから肯定応答信号を受け取ったわけではない場合、クラスタはその更新を中断し、取消し信号を出力することになる。一実施形態では、取消し信号によりノードが内部マッピング表現を再確認し、動作を続行する。

#### 【0084】

上記のように、クラスタの構成は、データを失わずにクラスタが動作している間に変更することができる。システム内のデータ・アクセス要求は遅延する可能性があるが、エラーなしで続行される。上記の再構成手順では、データを失わずに接続を再構成することもできる。たとえば、記憶装置はあるノードから切断して、他のノードに再接続することができる。物理的な再構成はステップ1222と1224の間で行うことができる。さらに、この再構成は、遅延を除き、クライアントにとってトランスペアレントなものである。上記の再構成の他の応用例としては、動作中にボリューム・マネージャのマッピング（または管理）を変更することがある。

#### 【0085】

上記の開示内容を完全に理解すると、当業者には多数の変形形態および修正形態が明らかになるだろう。特許請求の範囲はこのような変形形態および修正形態

をすべて包含するものと解釈することを意図するものである。

【図面の簡単な説明】

【図 1】

本発明の一実施形態によるクラスタ構成のブロック図である。

【図 2】

本発明の一実施形態による代替クラスタ構成のブロック図である。

【図 3】

本発明の一実施形態によるクラスタの 2 つのノード上で動作する仮想ディスク・システムのブロック図である。

【図 4】

本発明の一実施形態によるネットディスク・ドライバの初期設定を示すブロック図である。

【図 5】

本発明の一実施形態によるクラスタ・トランスポート・インタフェースの初期設定を示すブロック図である。

【図 6 a】

本発明の一実施形態による仮想ディスク・システムの動作を示す流れ図である。

【図 6 b】

本発明の一実施形態による仮想ディスク・システムの動作を示す流れ図である。

【図 7】

本発明の一実施形態によるネットディスク・ドライバの開始を示す流れ図である。

【図 8】

本発明の一実施形態によるクラスタ・トランスポート・インタフェースの開始を示す流れ図である。

【図 9】

本発明の一実施形態によるクラスタ・トランスポート・インタフェースのプロ

ック図である。

【図１０】

本発明の一実施形態による許可データを示す図である。

【図１１】

本発明の一実施形態による一貫した許可データの記憶およびアクセスを示す流れ図である。

【図１２】

本発明の一実施形態による構成マッピングの更新を示す流れ図である。

【図 1】

100

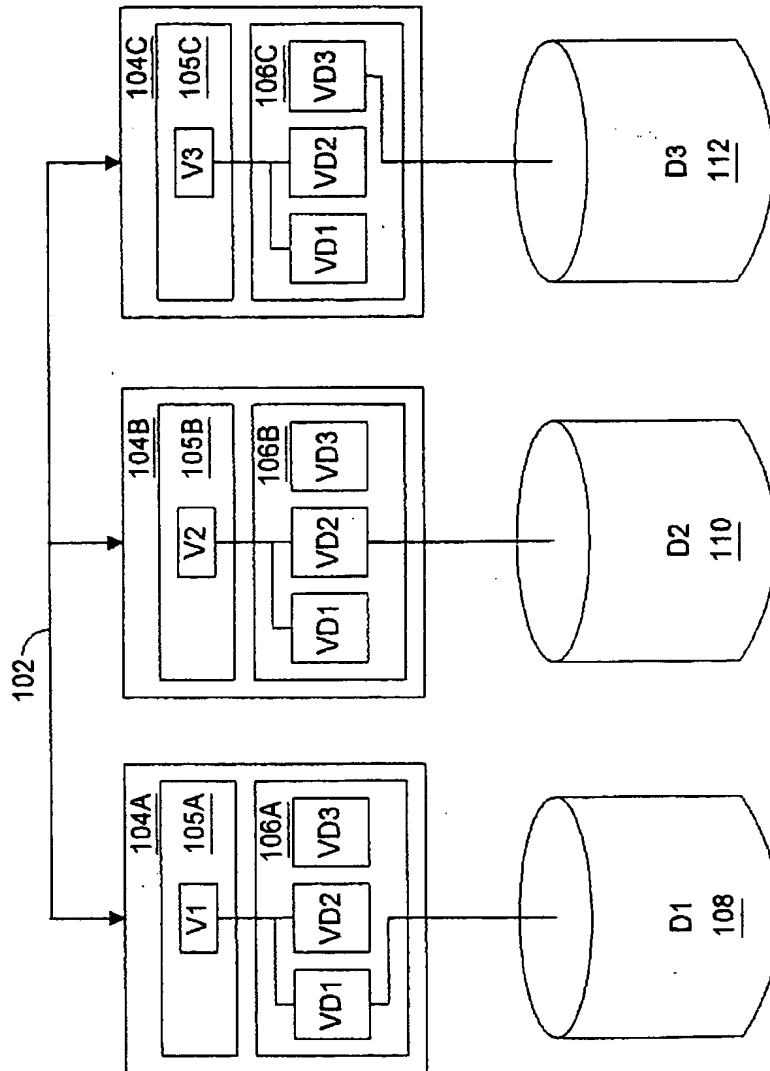


FIG. 1

【図 2】

200

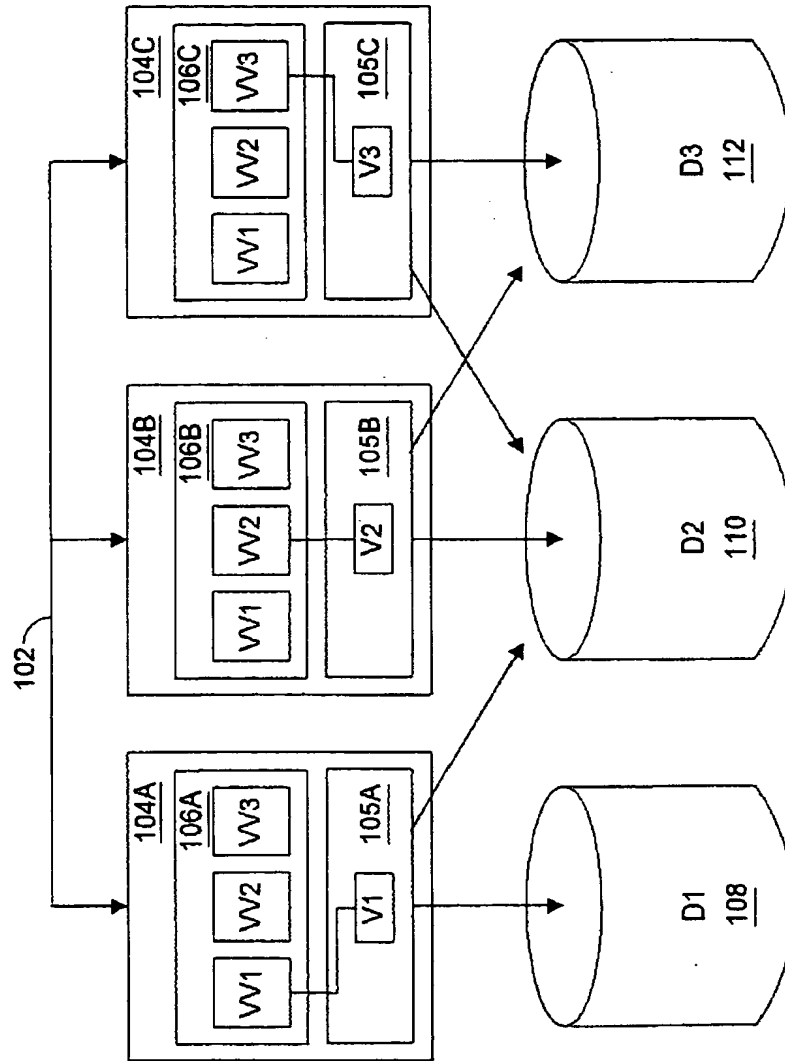


FIG. 2

Figure 1 is a block diagram illustrating a system architecture, divided into two main sections, 104A and 104B, connected via an INTERCONNECT 112.

**Section 104A (Left):**

- USER Section:** Contains CMM 310A, CCD 311A, CTID 316A, NDD 314A, and CLIENT 312A.
- KERNEL Section:** Contains CTI 322A, NETWORK TRANSPORT 328A, CCM 324A, ND 318A, NM 320A, and DISK DRIVER 326A.

**Section 104B (Right):**

- USER Section:** Contains CMM 310B, CCD 311B, CTID 316B, NDD 314B, and CTI 322B.
- KERNEL Section:** Contains NETWORK TRANSPORT 328B, CCM 324B, ND 318B, NM 320B, and DISK DRIVER 326B.

**Interconnect and Connections:**

- The INTERCONNECT 112 connects the CCM 324A of 104A to the CCM 324B of 104B.
- Arrows indicate data flow between components within each section and to/from disk devices.
- A dashed line separates the USER and KERNEL sections in each.

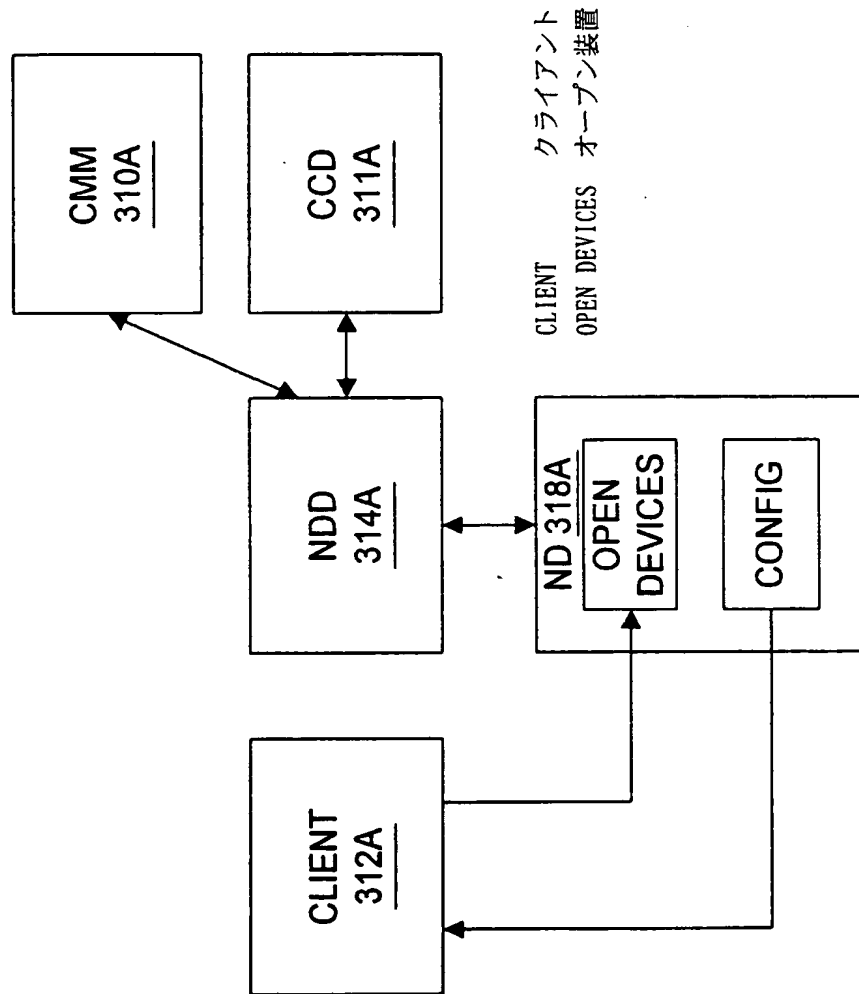
**Legend:**

- 1 1 2 相互接続
- 3 1 2 A クライアント
- 3 2 6 A ディスク・ドライバ
- 3 2 6 B ディスク・ドライバ
- 3 2 8 A ネットワーク・トラ
- 3 2 8 B ネットワーク・トラ
- 3 2 8 B ネットワーク・トラ

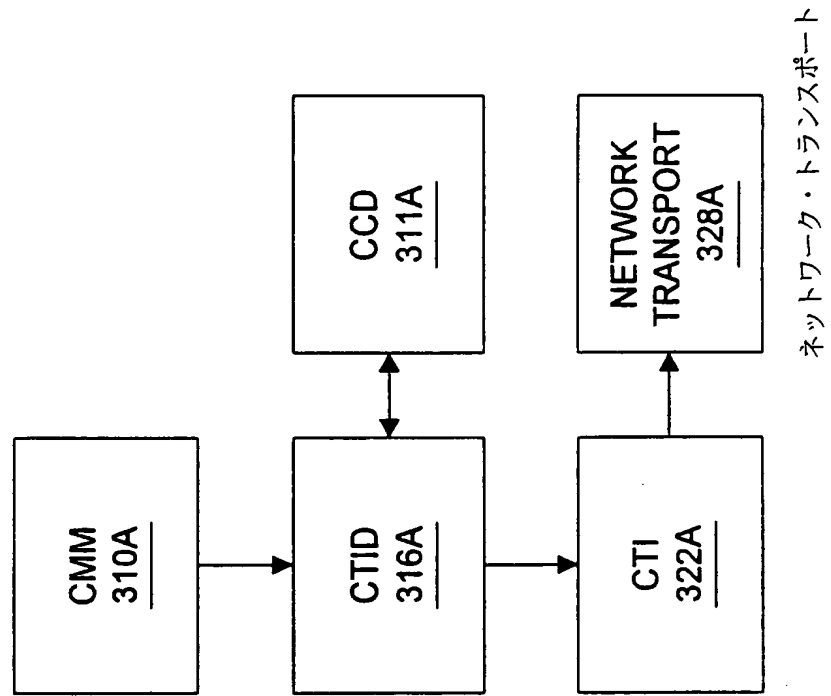
**Labels:**

- USER ユーザ
- KERNEL カーネル
- TO DISK DEVICE ディスク装置へ

【図 4】

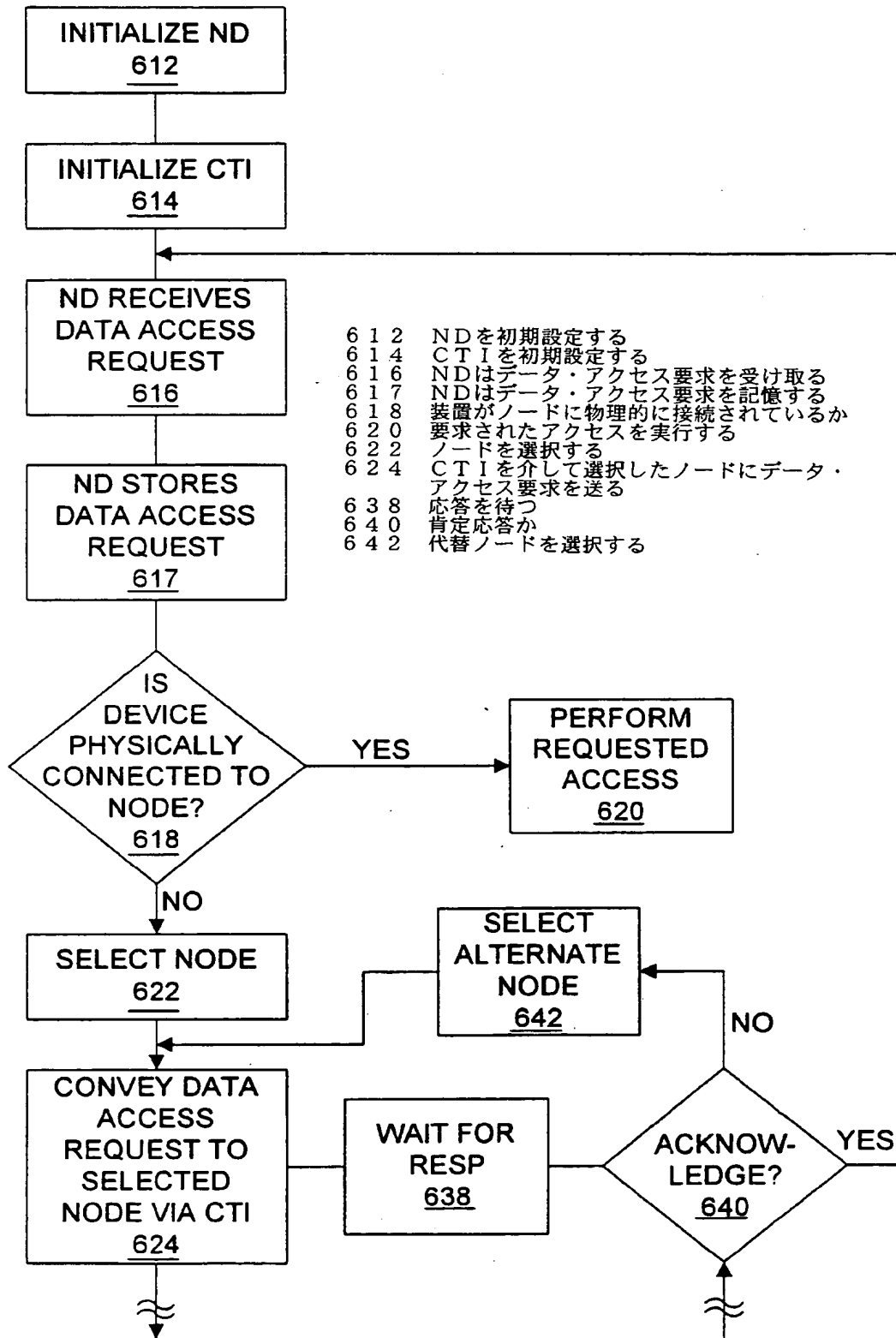


【図5】

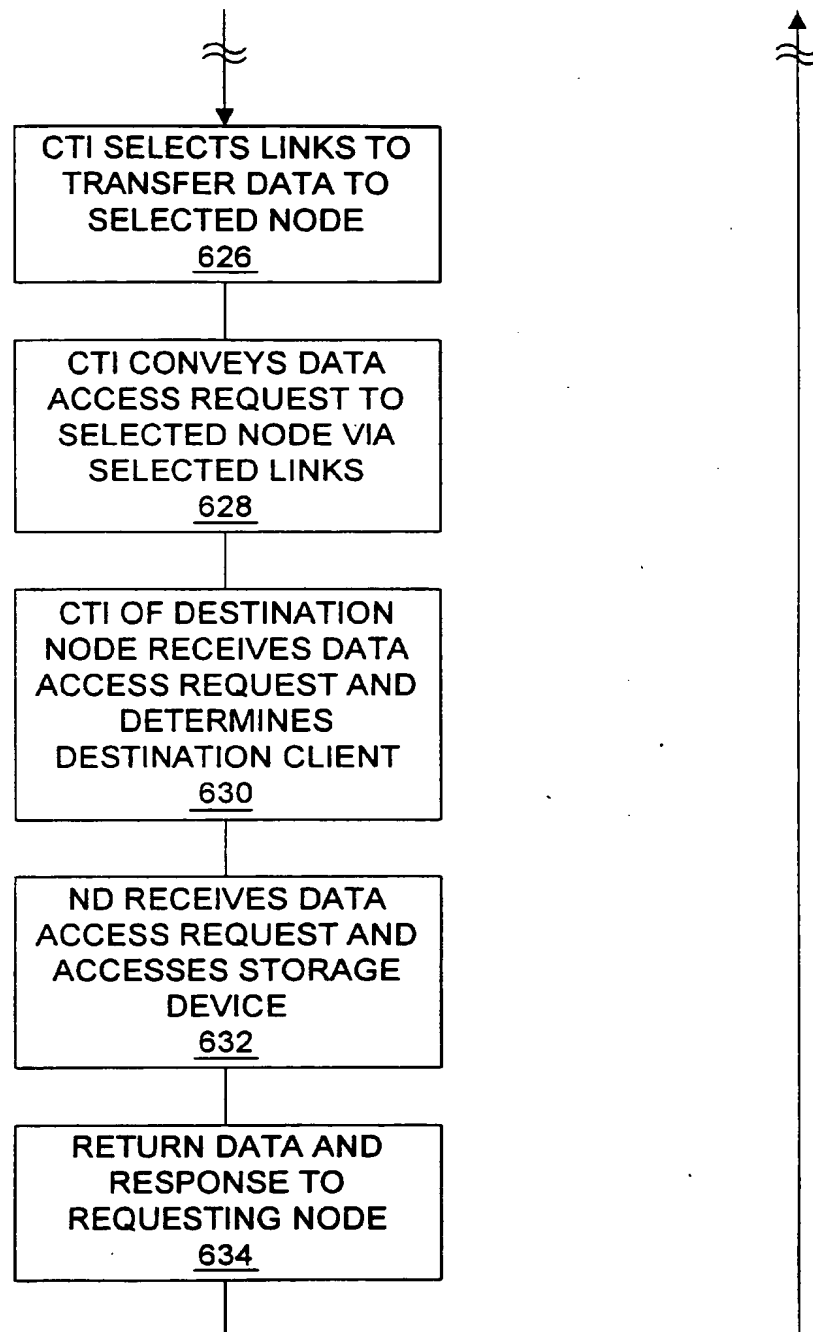




【図 6 a】

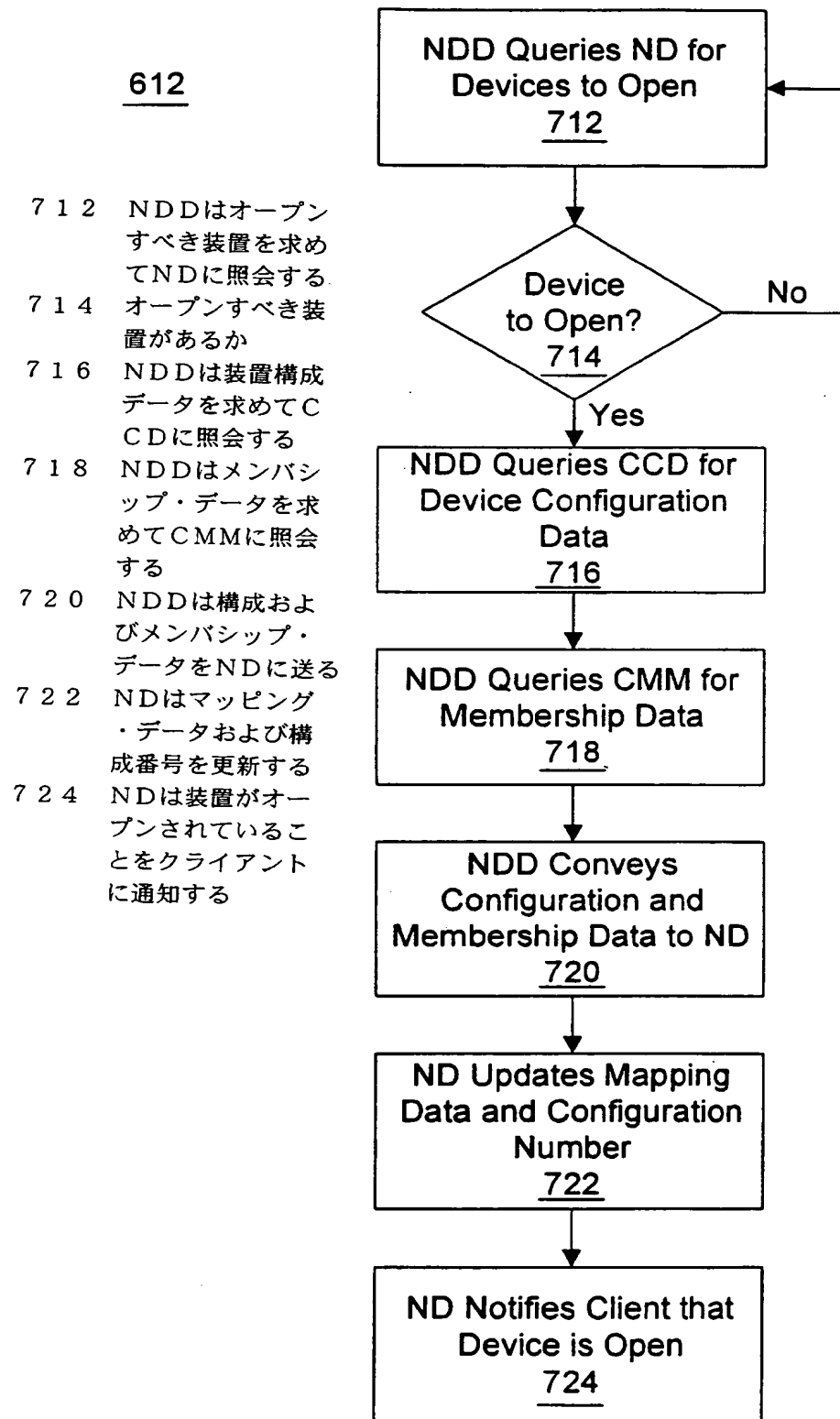


【図6b】



626 CTIは選択したノードにデータを転送するためのリンクを選択する  
 628 CTIは選択したリンクを介して選択したノードにデータ・アクセス要求を送る  
 630 宛先ノードのCTIはデータ・アクセス要求を受け取り、宛先クライアントを決定する  
 632 NDはデータ・アクセス要求を受け取り、記憶装置にアクセスする  
 634 要求側ノードにデータと応答を返す

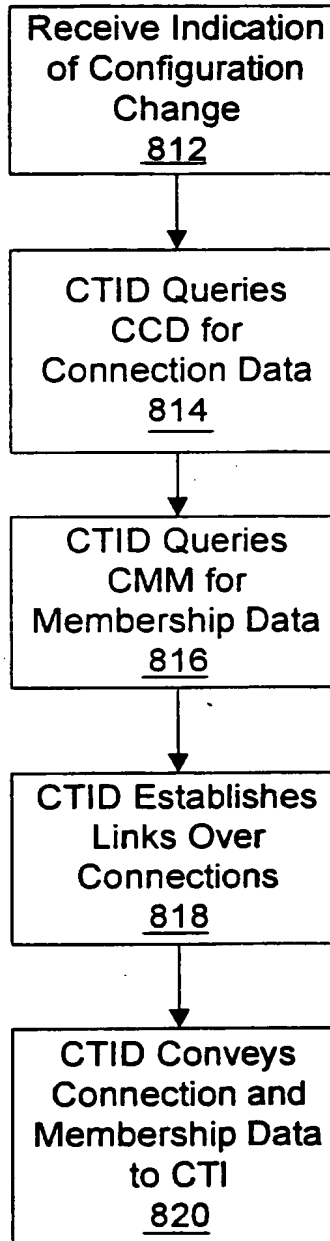
【図7】



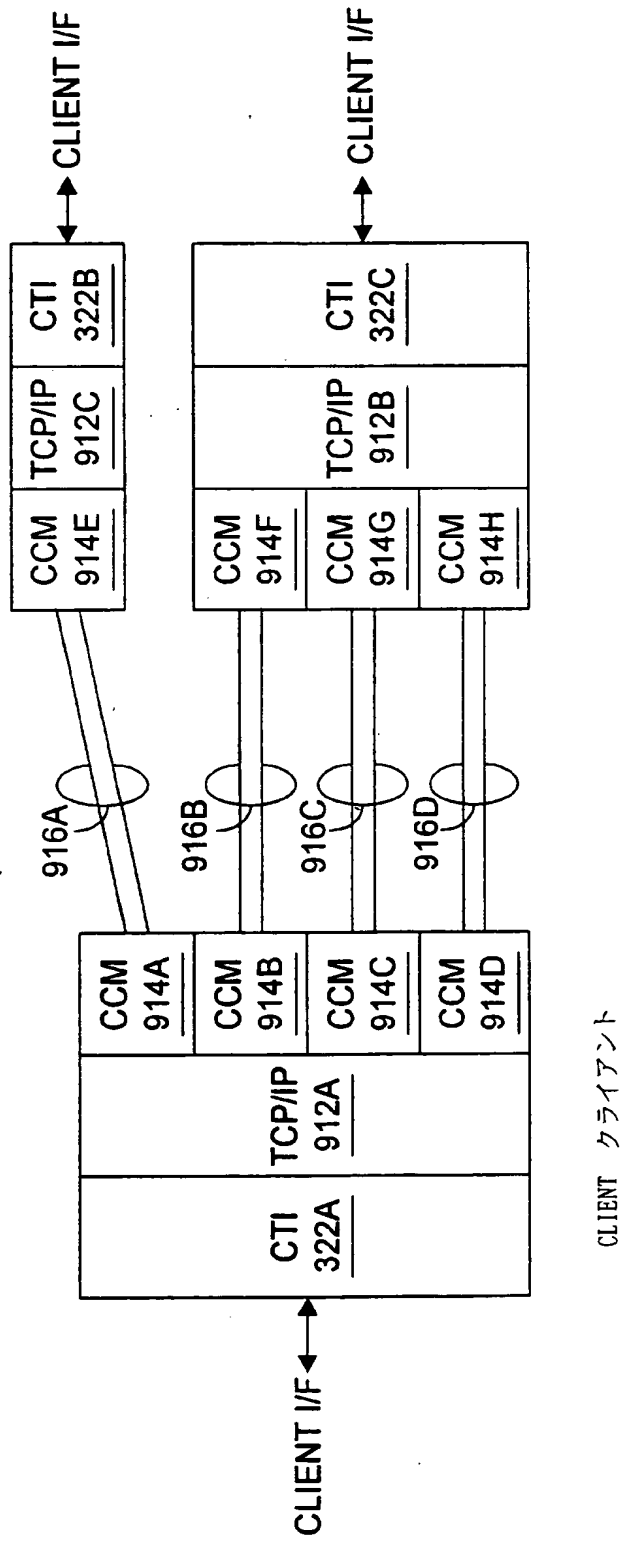
【図8】

614

- 812 構成変更の表示を受け取る  
814 CTIDは接続データを求めてCCDに照会する  
816 CTIDはメンバシップ・データを求めてCMMに照会する  
818 CTIDは接続によりリンクを確立する  
820 CTIDは接続およびメンバシップ・データをCTIに送る



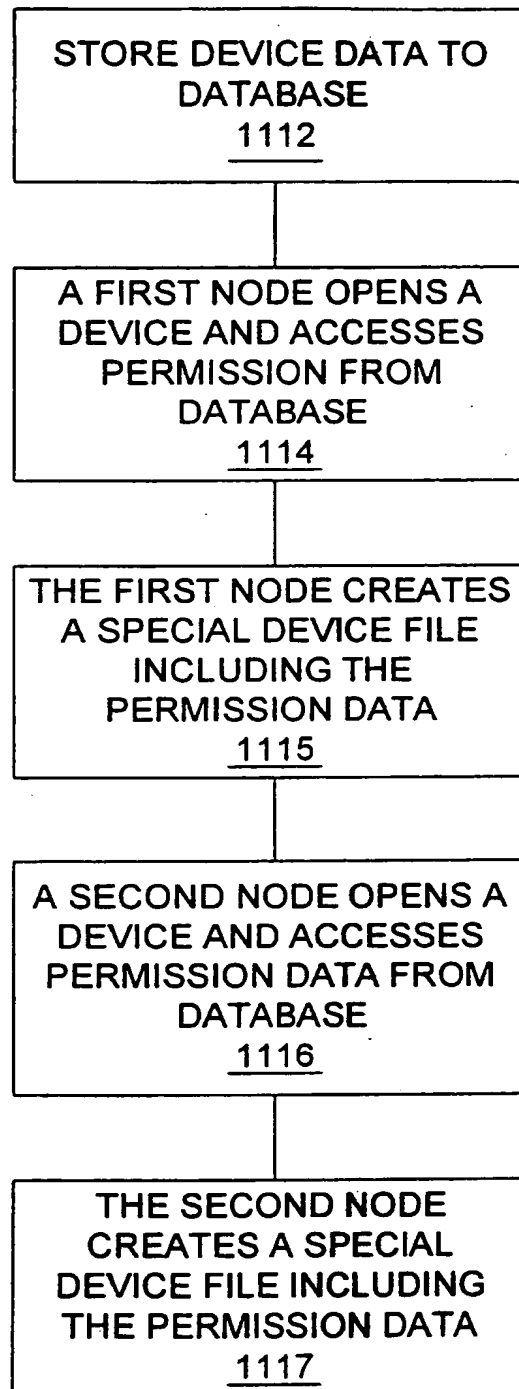
【図9】



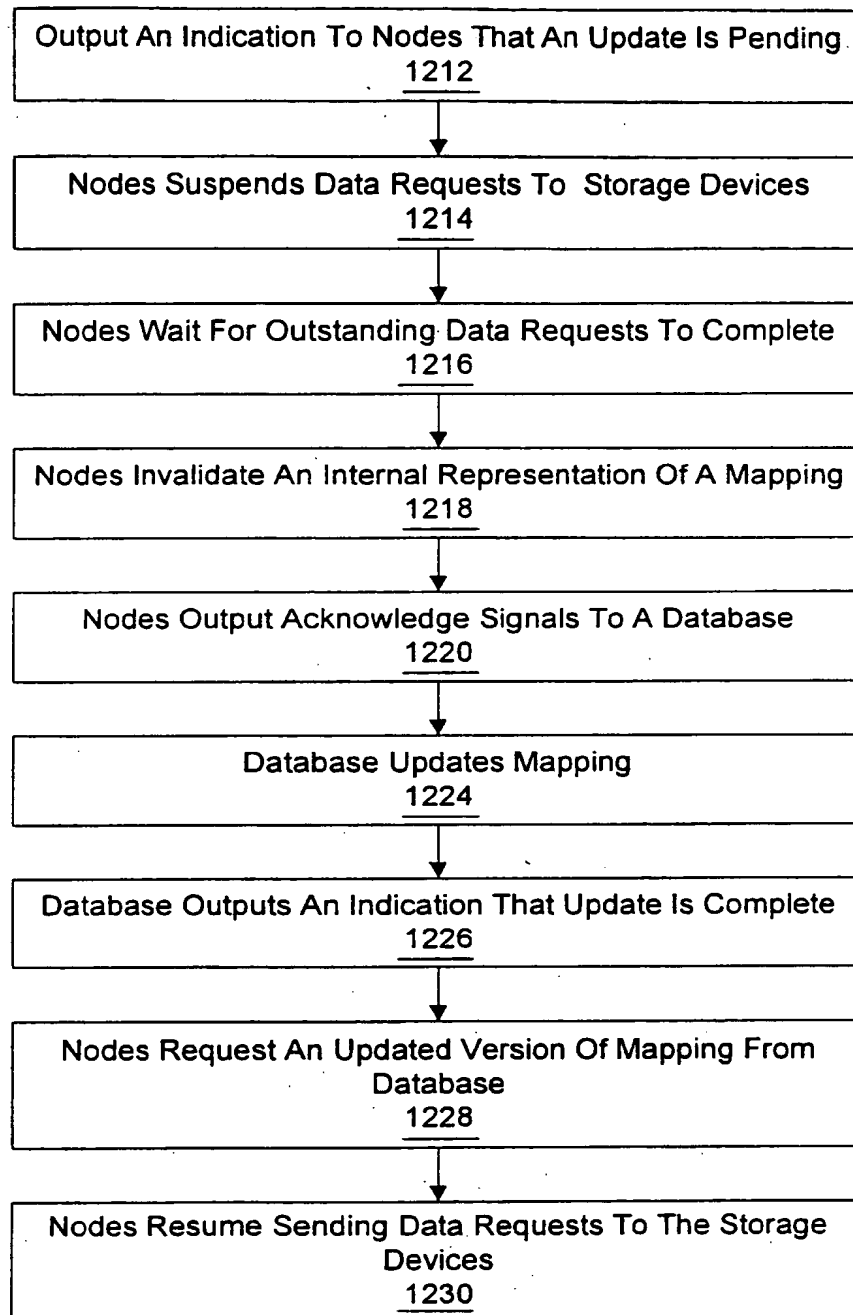


【図11】

- 1112 装置データをデータベースに記憶する
- 1114 第1のノードは装置をオープンし、データベースからの許可にアクセスする
- 1115 第1のノードは許可データを含む特殊装置ファイルを作成する
- 1116 第2のノードは装置をオープンし、データベースからの許可データにアクセスする
- 1117 第2のノードは許可データを含む特殊装置ファイルを作成する



【図12】



- |         |                                  |
|---------|----------------------------------|
| 1 2 1 2 | 更新が保留中であるという表示をノードに出力する          |
| 1 2 1 4 | ノードは記憶装置へのデータ要求を中断する             |
| 1 2 1 6 | ノードは未処理のデータ要求が完了するのを待つ           |
| 1 2 1 8 | ノードはマッピングの内部表現を無効にする             |
| 1 2 2 0 | ノードは肯定応答信号をデータベースに出力する           |
| 1 2 2 4 | データベースはマッピングを更新する                |
| 1 2 2 6 | データベースは更新が完了したという表示を出力する         |
| 1 2 2 8 | ノードはマッピングの更新済みバージョンをデータベースから要求する |
| 1 2 3 0 | ノードは記憶装置へのデータ要求の送信を再開する          |



【手続補正書】特許協力条約第34条補正の翻訳文提出書

【提出日】平成12年6月26日(2000.6.26)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正内容】

【特許請求の範囲】

【請求項1】 通信リンクに結合された第1のノードと、第2のノードと、第3のノードと、

第2および第3のノードに結合され、データを記憶するように構成された記憶装置とを含み、

前記第1のノードは仮想ディスク・システムを使用して前記記憶装置にアクセスし、前記記憶装置にアクセスする一次ノードと前記記憶装置にアクセスする代替ノードを識別するマッピング・データを記憶するように構成され、

前記第1のノードは前記通信リンクを介してデータ・アクセス要求を発行するドライバを含み、前記第2および第3のノードはそれぞれ、前記記憶装置からのデータにアクセスし、前記通信リンクを介して応答を送るように構成されたマスタを含み、前記ドライバは前記マッピング・データによって識別された前記一次ノードにデータ・アクセス要求を送るように構成される、分散コンピューティング・システム。

【請求項2】 前記ドライバは前記ドライバが前記応答を受け取るまで前記データ・アクセス要求のコピーを記憶するように構成される請求項1に記載の分散コンピューティング・システム。

【請求項3】 前記ドライバは前記ドライバが前記一次ノードからの応答を受け取り損なった場合にマッピング・データによって識別された前記代替ノードに前記データ・アクセス要求を再送するようにさらに構成される請求項1または2に記載の分散コンピューティング・システム。

【請求項4】 前記一次ノードが前記第2および第3のノードのうち的一方

であり、前記代替ノードが前記第2および第3のノードのうちのもう一方である請求項1ないし3のいずれかに記載の分散コンピューティング・システム。

【請求項5】 前記マッピング・データが前記第2のノードを一次ノードとして識別し、前記マッピング・データが前記第3のノードを代替ノードとして識別する請求項1ないし4のいずれかに記載の分散コンピューティング・システム。

【請求項6】 前記第1のノードはアクティブ・ノードのリストを含むメンバシップ・データを記憶するようにさらに構成される請求項1ないし5のいずれかに記載の分散コンピューティング・システム。

【請求項7】 前記アクティブ・ノードのリストが前記一次ノードを含まない場合、前記ドライバは前記代替ノードにデータ・アクセス要求を送るように構成される請求項6に記載の分散コンピューティング・システム。

【請求項8】 前記第1、第2、および第3のノードが前記マッピング・データの同一コピーを記憶する請求項1ないし7のいずれかに記載の分散コンピューティング・システム。

【請求項9】 前記第1、第2、および第3のノードが前記マッピング・データを含むクラスタ構成データベースの同一コピーを記憶するように構成される請求項8に記載の分散コンピューティング・システム。

【請求項10】 前記第1、第2、または第3のノードがインアクティブになったときに前記マッピング・データが更新される請求項1ないし9のいずれかに記載の分散コンピューティング・システム。

【請求項11】 前記ドライバは、前記マッピング・データが更新されたときに前記データ・アクセス要求の送信を中断するように構成される請求項1ないし10のいずれかに記載の分散コンピューティング・システム。

## 【国際調査報告】

## INTERNATIONAL SEARCH REPORT

International Application No. <b>PCT/US 99/09903</b>	
<b>A. CLASSIFICATION OF SUBJECT MATTER</b> IPC 6 G06F11/14	
According to International Patent Classification (IPC) or to both national classification and IPC	
<b>B. FIELDS SEARCHED</b> Minimum documentation searched (classification system followed by classification symbols) IPC 6 G06F	
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched	
Electronic data base consulted during the international search (name of data base and, where practical, search terms used)	
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>	
Category *	Citation of document, with indication, where appropriate, of the relevant passages Relevant to claim No.
X	EP 0 709 779 A (INTERNATIONAL BUSINESS MACHINES) 1 May 1996 (1996-05-01) column 5, line 18 - column 6, line 35
A	US 5 475 813 A (INTERNATIONAL BUSINESS MACHINES) 12 December 1995 (1995-12-12) abstract
<input type="checkbox"/> Further documents are listed in the continuation of box C.	
<input checked="" type="checkbox"/> Patent family members are listed in annex.	
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (see specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "Z" document member of the same patent family	
Date of the actual completion of the international search <b>27 September 1999</b>	Date of mailing of the international search report <b>04/10/1999</b>
Name and mailing address of the ISA European Patent Office, P.B. 5610 Patentlaan 2 NL - 2260 HV Rijswijk Tel. (+31-70) 840-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer <b>Corremans, G</b>

Form PCT/ISA/210 (second sheet) (July 1998)

## INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 99/09903

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
EP 709779	A	01-05-1996	JP 8255122 A	01-10-1996
			US 5668943 A	16-09-1997
US 5475813	A	12-12-1995	NONE	

## フロントページの続き

(51)Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード* (参考)
G 0 6 F 13/10	3 4 0	G 0 6 F 13/10	3 4 0 B
(31)優先権主張番号	0 9 / 0 7 6 , 3 4 6		
(32)優先日	平成10年5月12日(1998. 5. 12)		
(33)優先権主張国	米国 (U S)		
(31)優先権主張番号	0 9 / 0 7 6 , 2 7 4		
(32)優先日	平成10年5月12日(1998. 5. 12)		
(33)優先権主張国	米国 (U S)		
(81)指定国	EP(AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG), AP(GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW		
(71)出願人	901 SAN ANTONIO ROAD PALO ALTO, CA 94303, U. S. A.		
(72)発明者	トラバーサット, ベルナルド・エイ アメリカ合衆国・94109・カリフォルニア州・サン フランシスコ・カリフォルニア ストリート・2055・アパートメント 402		
(72)発明者	ハーンドン, ロバート アメリカ合衆国・80906・コロラド州・コロラド スプリングス・サウス ネバダ アベニュー 137番・1837		
(72)発明者	ジェン, シャオヤン アメリカ合衆国・94555・カリフォルニア州・フレモント・ゴルービン コモン・ 5454		
(72)発明者	ブロック, ロバート・ジェイ アメリカ合衆国・94043・カリフォルニア州・マウンテン ビュー・ノース レング ストーフ 29番・265		

F ターム(参考) 5B014 EA04 HC02 HC15  
5B027 AA00 BB06 BB07  
5B034 BB11 CC05  
5B082 DA01 DE01 DE02 DE04 FA07  
HA05 JA01  
5B083 AA09 BB03 CD06 CE01 DD13  
EE11 GG04

【要約の続き】

実行され、マッピングが変更される。次にそのノードはマッピングの内部表現を更新し、データ・アクセス要求の発行を再開する。